A CRIMINAL PERSPECTIVE ON EXPLOIT PACKS

Introduction

Criminals refer to it as a “BEP”. As the name intones the entire computer exploit process begins with a web browser. The web browser’s ubiquitous use in daily life has given rise to the Browser Exploit Pack as the infection vehicle darling of the Underground.

There are multiple Internet channels available for pushing malware to a victim such as email, P2P file sharing, instant messenger, and social media. A live exploit pack only requires a victim “drive by” – a trivial website visit – a soft push after exploring layer 7 for vulnerabilities.

We installed and configured over 40 exploit packs[1] in order to better understand the different family’s value in criminal use scenarios. In this paper we chronicle the exploit pack genesis and historical evolution. We discuss the spectrum of technical acumen required to successfully install and use different exploit pack families. Finally we detail the monetization and code protection mechanisms currently in place as well as the overall effectiveness of these different exploit pack families.

For optimal exploration we created dedicated networks for exploit pack installation. Our drive by victim machines were running Windows XPThe client machine acting as a drive by victim was running Windows XP SP2 and we excluded all further patches[2]. To further give these exploit packs every chance of exploitation success we installed old versions of Internet Explorer, Firefox, Opera, Adobe Reader, Flash, Java Virtual Machine, Windows Media Player, and other applications. Generally the application version was matched to a release in late 2004 or early 2005 since that was the approximate time frame that the first exploit packs were released.

---

[1] A special thank you to the security community for their assistance in locating/sharing various exploit packs

[2] This was a reasonable operating system state since Windows XP SP2 use is still pervasive.

Evolution

We remember the first exploit packs being publicly released in 2005. Prior to that most exploit capable malware was written in C++ and used in IRC. One of the first available sources was SDBot in 2002, and similar exploit capable IRC bots were released in the following years. In the beginning IRC bots used exploit techniques that primarily focused on services scanning[3] and using vulnerabilities for those discovered services. Although scanning is still prevalent, Microsoft's introduction of the Windows XP SP2 default enabled firewall led to decreased scanning effectiveness and resulting lower numbers of mass infections.

Early IRC bot's update channels often relied on HTTP. Logically, HTTP quickly became the criminal control protocol of choice and has long since eclipsed IRC. The earliest exploit pack families like Mpack, Icepack, and Infector have evolved into newer versions (Icepack Platinum) and newer families (Black Hole, Incognito, Bleeding Life). In fact, most of the exploit pack families have undergone multiple version updates in the past 6 six years. We tested 5 versions of 0x88, 3 versions of Eleonore, 4 versions of El Fiesta, 5 versions of Icepack, 3 versions of Fragus, 2 versions of NeoSploit, 6 versions of Mpack, etc.

Typically the version updates we examined contained (naturally) additional exploits, but many also contained bug fixes and more intuitive and visually appealing control panel interfaces through the inclusion of AJAX[4] functionality and larger images. For example, Black Hole is currently very popular[5], but the exploits it uses are almost identical to many other packs, likely indicating that the control panel – a tasteful and well-designed user interface that incorporates professional CSS, crisp icons, and Javascript - is responsible for its prolific Underground status.

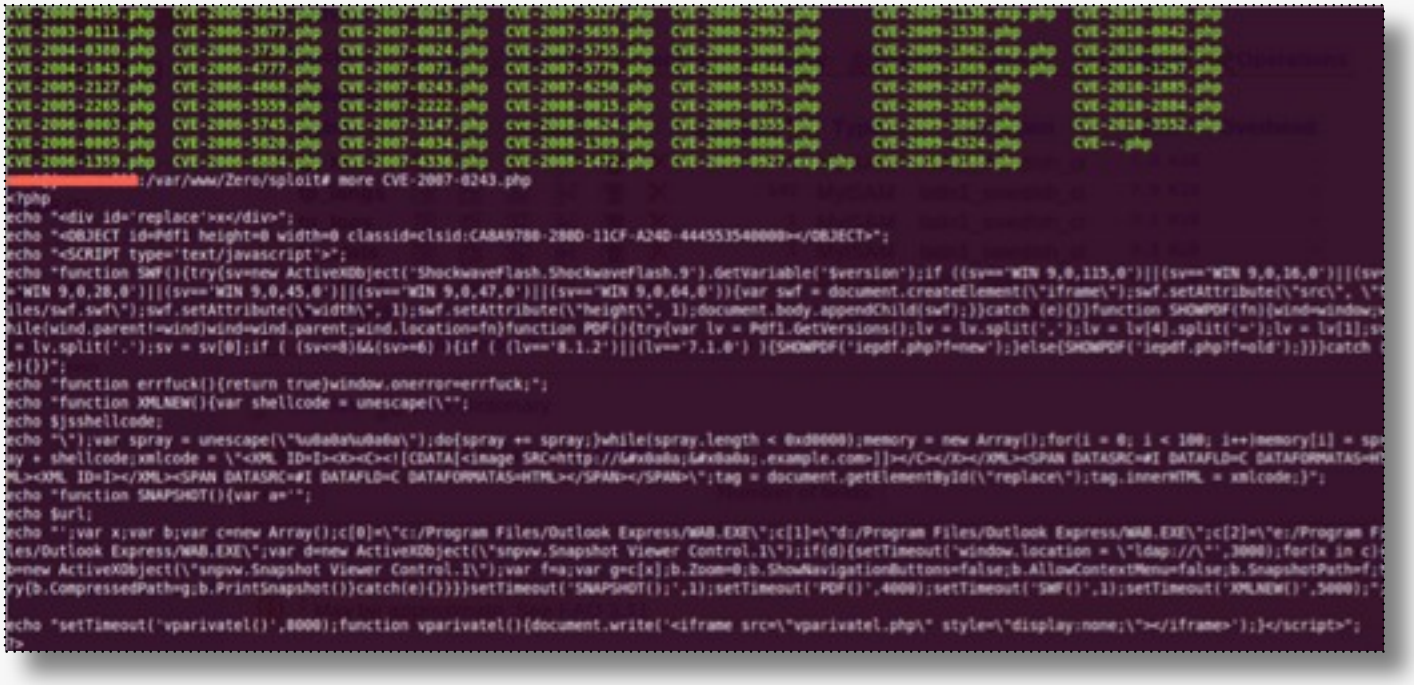
[3] Agobot, Spybot, rBot etc.

[4] Asynchronous JavaScript and XML

[5] http://www.eweek.com/c/a/Security/Blackhole-Exploit-Kit-Behind-USPS-Attack-224683/

Installation

Essentially a browser exploit pack is a bundle of PHP and HTML files coupled with a list of exploit files (sometimes named by CVE[6]) designed to target the OS, browser, or other application. Once installed, the exploit pack provides an easy to use control page for the exploit process, accessible to anyone with basic Internet experience.

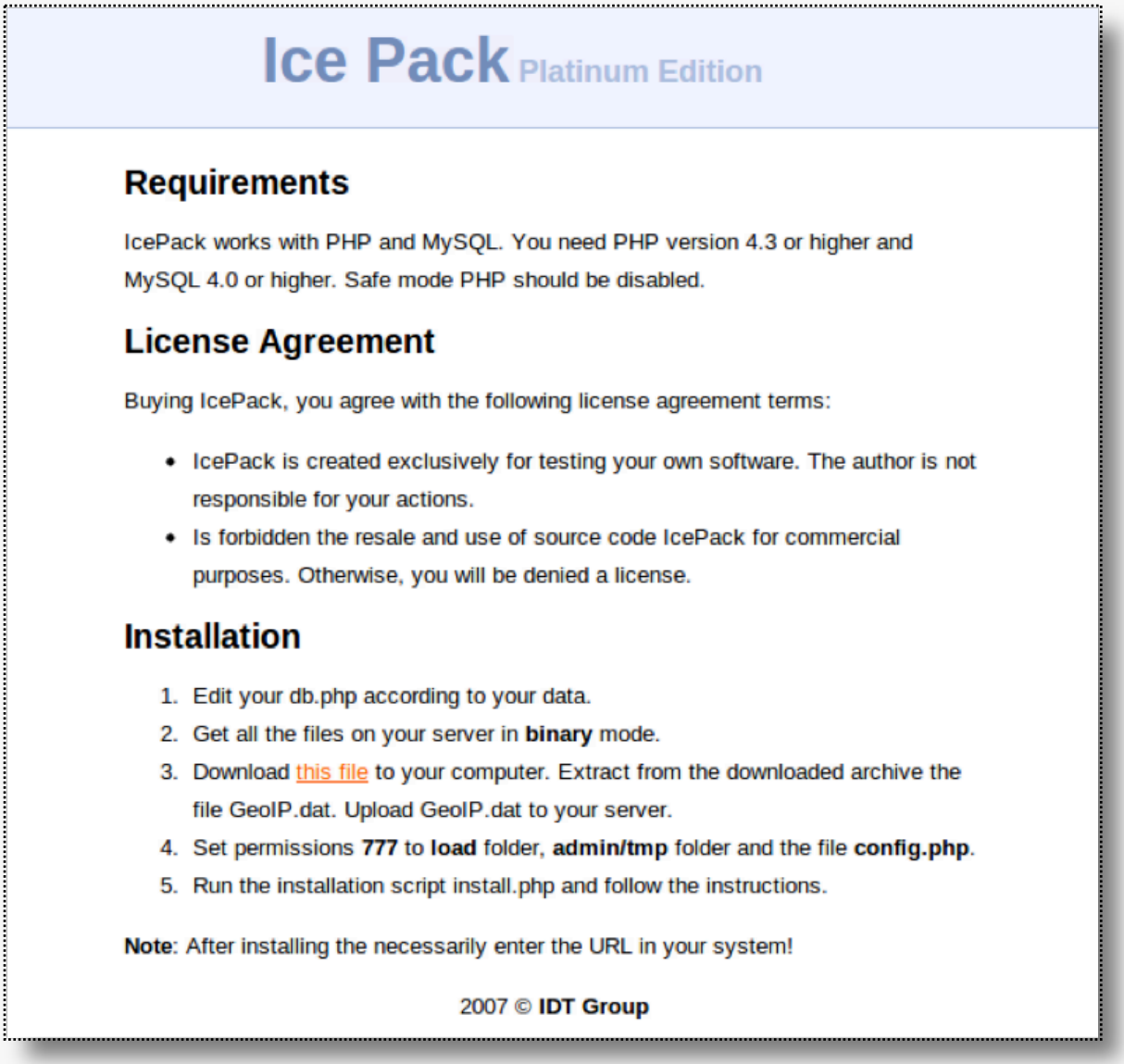


Listing of several exploit codebase in Zero exploit pack

Newer generations contained additional JavaScript and Cascading Style Sheet (CSS) files for an enhanced control panel experience, but fundamentally all of these packs were similar. Even the directory structure deviated only slightly between families.

[6] The Common Vulnerabilities and Exposures framework : cve.mitre.org

When we began installing we found that few of the exploit packs were accompanied by installation instructions. Crimepack and Icepack were two exceptions that included reasonable documentation.



Ice Pack Platinum Edition

Requirements

IcePack works with PHP and MySQL. You need PHP version 4.3 or higher and MySQL 4.0 or higher. Safe mode PHP should be disabled.

License Agreement

Buying IcePack, you agree with the following license agreement terms:

- IcePack is created exclusively for testing your own software. The author is not responsible for your actions.
- Is forbidden the resale and use of source code IcePack for commercial purposes. Otherwise, you will be denied a license.

Installation

1. Edit your db.php according to your data.
2. Get all the files on your server in **binary** mode.
3. Download this file to your computer. Extract from the downloaded archive the file GeoIP.dat. Upload GeoIP.dat to your server.
4. Set permissions **777** to **load** folder, **admin/tmp** folder and the file **config.php**.
5. Run the installation script install.php and follow the instructions.

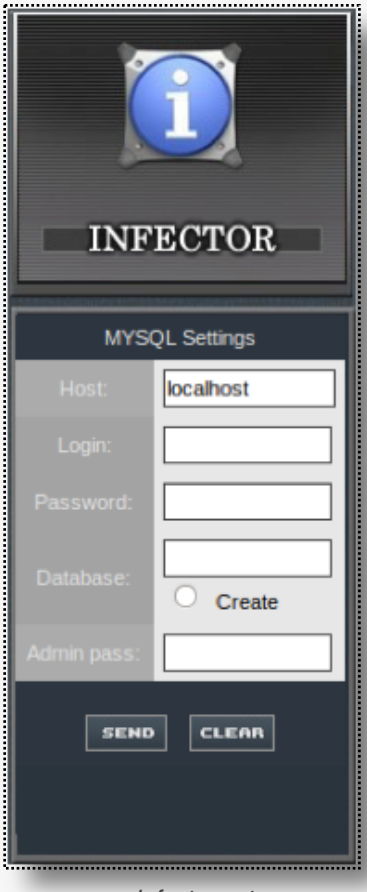
Note: After installing the necessarily enter the URL in your system!

2007 © **IDT Group**

Install Manual included with Ice Pack

Every exploit pack we tested relied on LAMP[7]. Successful setup required a basic knowledge of Apache, PHP installation, MySQL, and a general familiarity with UNIX. The "Infector exploit pack" (2006) contained a GUI option on the setup.php page for database creation, but it failed to create our database.

[7] Linux/Apache/Mysql/PHP



Infector setup

Almost every exploit pack we configured required a MySQL database. When a database layout wasn't included with the source, we were left guessing about the specific database structure. As most databases looked similar and the PHP code provided clues, we were ususally successful in rebuilding the databases to get the code to work.

The Unique exploit pack provided a well-organized and straight forward config.php file for variable configuration. All of the exploit packs contained a config.php or admin.php file which was almost always located in the root directory or in the "admin" directory. In addition to specifying the correct path for the exploit redirection PHP files, other important variables included credentials for the database and administrator control panel as well as the databases' location. The administrator page was usually found in /admin/admin.php, /admin/stats.php, or /stats.php. The older exploit packs contained fewer directories and a longer list of files in the root directory.

The newer exploit kits referenced control panel password hashes that required editing where the password was not explicitly provided. Generally the hash was created from a typical SQL password

```
    I A  config.php (php)  <?
<?

/*****************************************\
|*.........Made.by.securlty.&.mazd.........*|
|*.....You.have.made.a.correct.choice!.....*|
|*............Sheaf.Sploit.Pack............*|
\*****************************************/

// ALL
$url='http://domain.com/folder/getexe.php';      // Link to getexe.php
$pdf='http://domain.com/folder/pdf.php';         // Link to pdf.php
$noun='404';                                     // Redirect non-unique; '404' to old-style 404 error;
define('EXE', 'load.exe');                       // Name for you exe file
define('Unique', 'go-go-go');                    // Don't edit

// FOR ACCESS MySQL
$db_host='localhost';                            // DB Host
$db_name='admin_un';                             // DB Name
$db_user='admin_un';                             // DB User
$db_pass='123456';                               // DB Pass

// FOR ACCESS ControlPanel
$cp_url='stats.php';                             // CP url
$cp_user='admin';                                // CP user, default login: admin
$cp_pass='35f504164d5a963d6a820e71614a4009';     // CP pass, default password: pass

?>
```

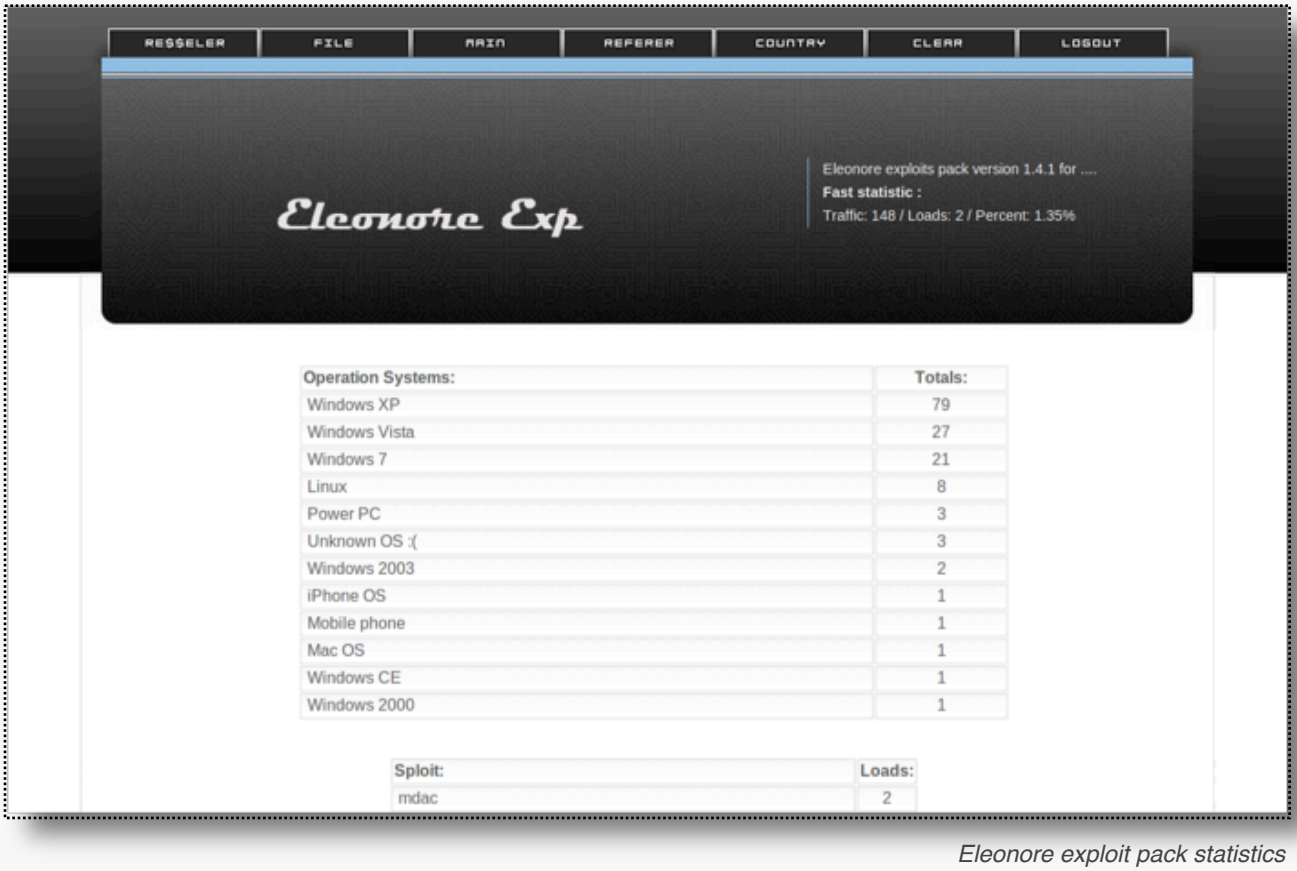
configuration file with admin username and password specified

Approximately half of the exploit packs we tested contained a .sql file for importation, which naturally made the database configuration easier. Other exploit packs contained no help files for the expected database structure. We were left guessing. One table? How many fields? A fair amount of trial and error was required.

Conversely, Eleonore was a perfect experience with easy database setup. Once the .sql file was imported the control panel immediately displayed the statistics from the last criminal who created the file.

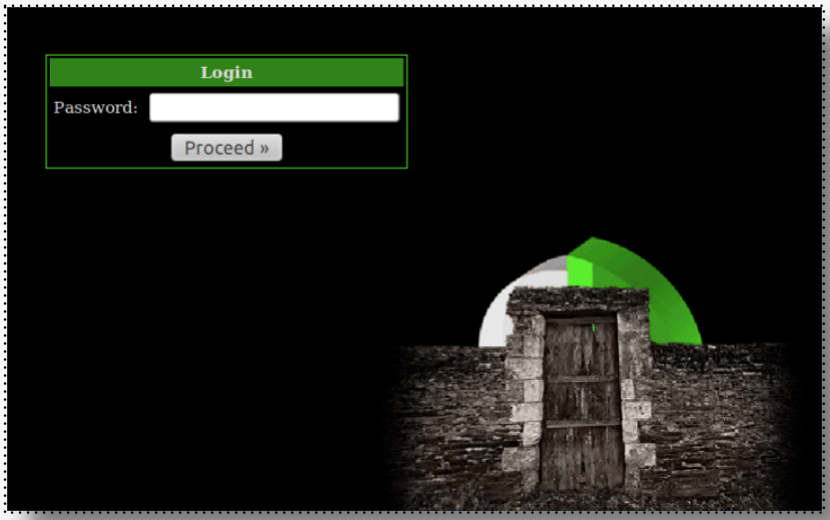
Browsers:	Traffic:	Loads:	Percent:
Chrome 4.0	2	0	0
Chrome 5.0	1	0	0
FireFox 2.0.0	1	0	0
FireFox 3.0.17	3	0	0
FireFox 3.0.3	5	0	0
FireFox 3.0.4	1	0	0
FireFox 3.0.7	1	0	0
FireFox 3.0.8	1	0	0
FireFox 3.5.2	4	0	0
FireFox 3.5.3	1	0	0
FireFox 3.5.5	1	0	0
FireFox 3.5.6	2	0	0
FireFox 3.5.7	21	0	0
FireFox 3.6	5	0	0
iPhone	1	0	0
Mobile phone browser	1	0	0
Mozilla 5.0	1	0	0
MSIE 6.0	8	2	25
MSIE 7.0	19	0	0
MSIE 8.0	26	0	0
Opera 9.50	1	0	0
Opera 9.63	2	0	0
Opera 9.64	9	0	0
Opera 9.80	24	0	0
Opera Mini/mobile phone	1	0	0
Power PC	3	0	0
Unknown browser :(	3	0	0
Total stats:	148	2	1.35 %

SQL file statistics visible after import



Eleonore exploit pack statistics

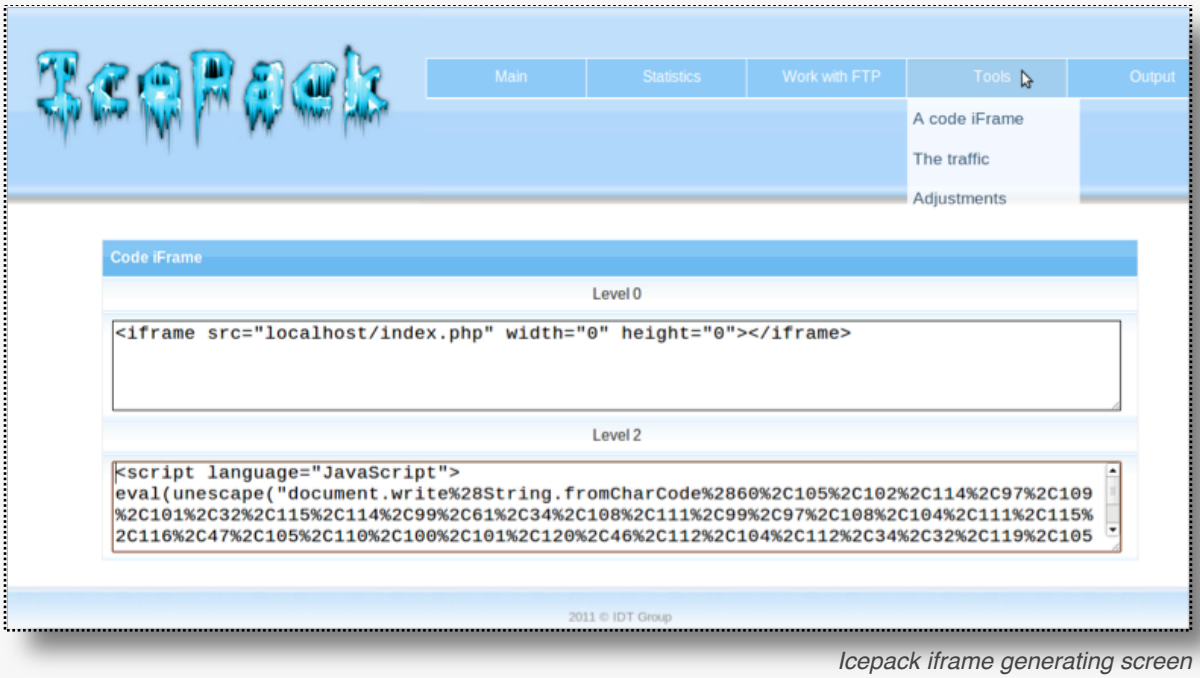
The administrator control panels were all very similar because they contained the same functionalities and for the most part changes in families or versions all appeared to be incremental copies of each other



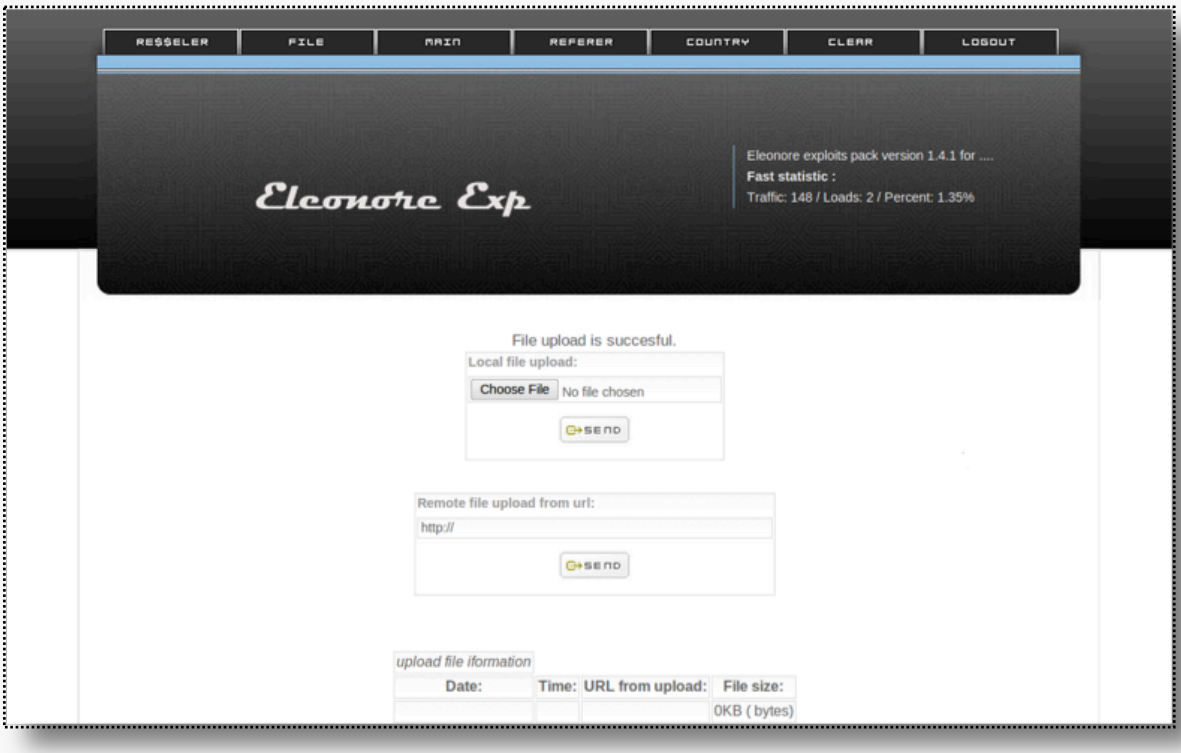
Sploit25 login screen with password only option

Value added control panel elements included JavaScript (a few included encoding) iframe generation, local or remote file upload for a malicious payload, and granular searching and file loading across the infected victim set.

Fragus automated javascript generating form



Icepack iframe generating screen



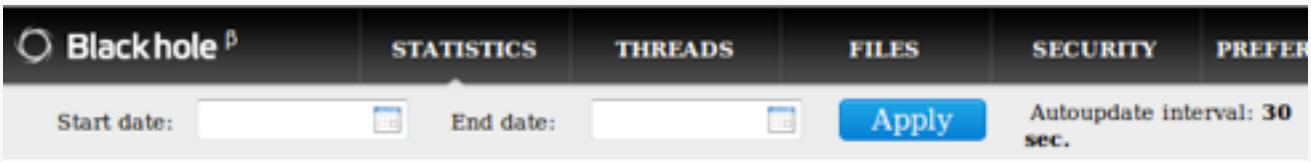
Eleonore upload screen

Eleonore iframe generating screen

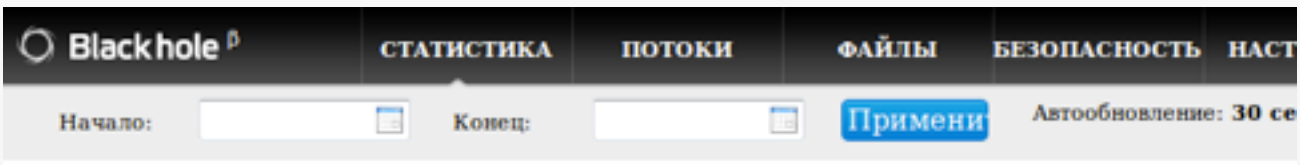
A number of the older exploit packs like Adpack, Gpack, Siberia, and Yes contained foreign character sets that made navigating the control panel difficult, and obviously they were built for a non-English speaking audience. The most prominent 'foreign' language we encountered was Russian. A few control panels even presented an English or Russian login option.

Language options in exploit kits

Black Hole was another prominent exploit pack built in Russian for the ever present Russian speaking exploit pack community. It was built with an English and Russian Language login option.

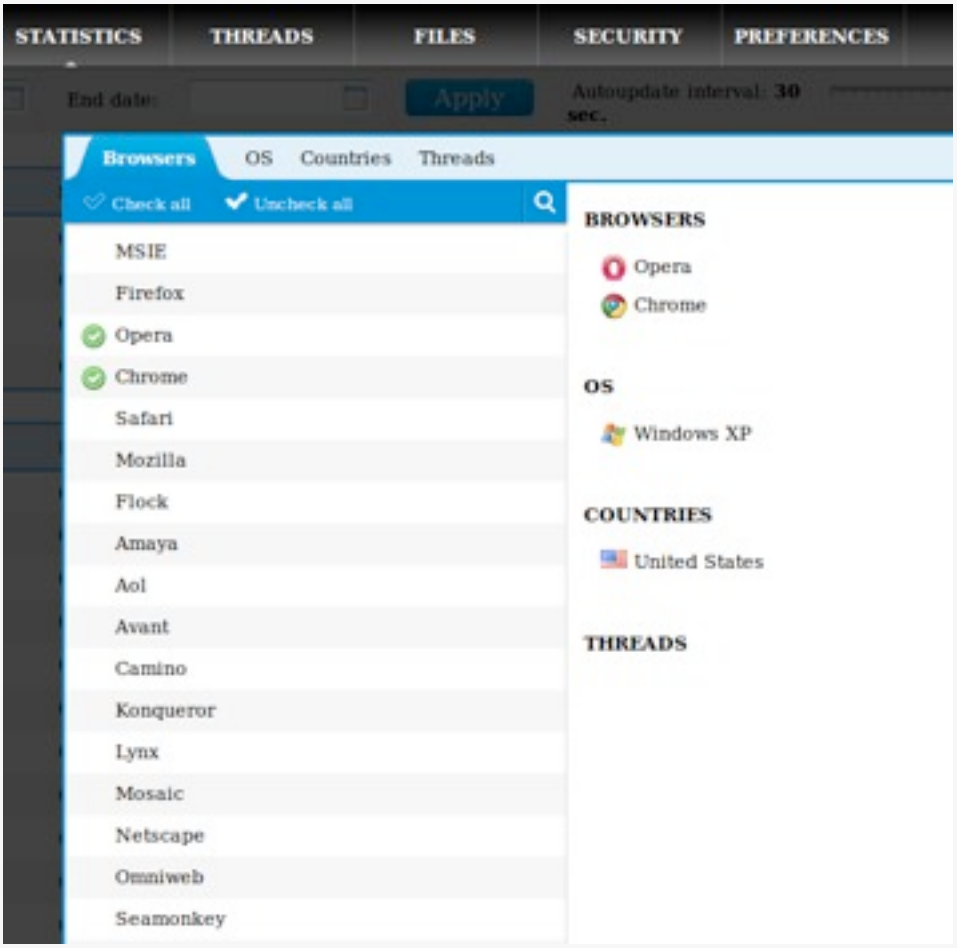


Blackhole with English menu

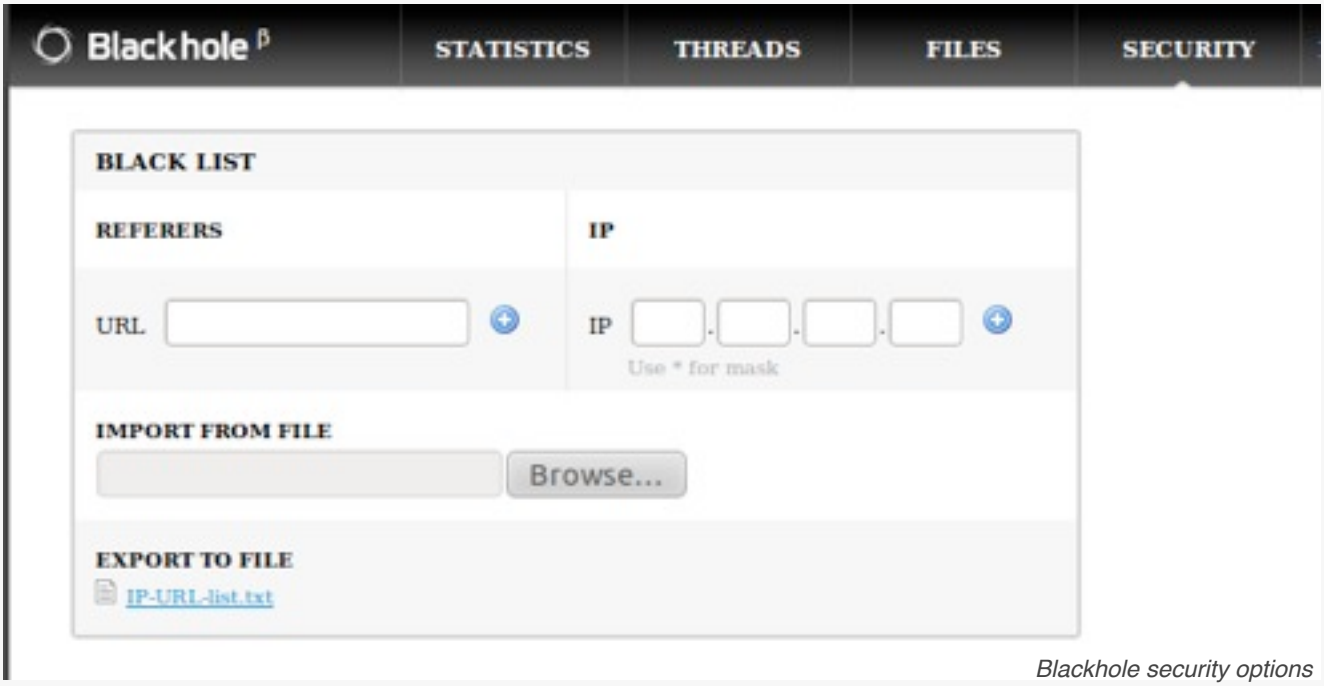


Blackhole with Russian menu

The Blackhole exploit pack presented a highly evolved user interface that extensively used dynamic AJAX page updates. One of the key updates was the user defined statistics widgets. The option to block specific HTTP referrers or IP addresses was also very intuitive.



Blackhole personalized widget option

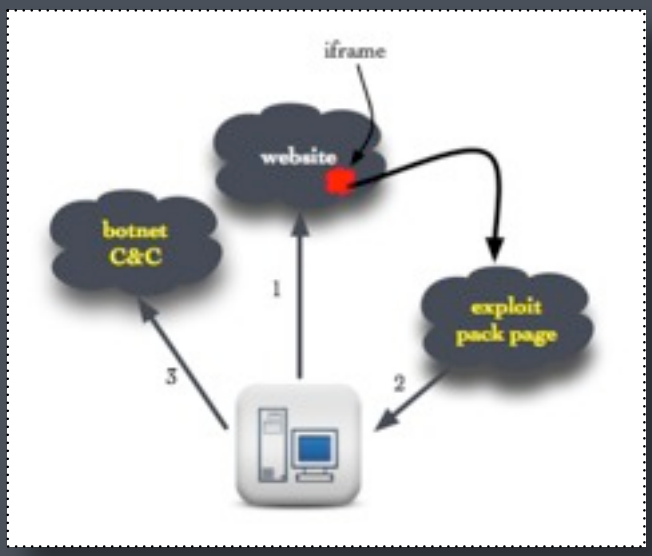


Blackhole security options

EFFECTIVENESS

Exploit packs are criminally appealing because victim exploitation is quick and seamless if a vulnerability exists. The victim may not notice anything different in their computer's behavior post drive-by. The majority of exploit packs we tested simply required the client computer visit the root index.php page. The newer families required a drive-by to a specifically crafted URI that corresponded to a specific "user" (renter) of the managed exploit pack.

The exploit pack itself resides squarely in the middle of a victim computer's journey to joining a multi-source malware HTTP botnet like SpyEye. The first step is typically HTTP redirection. A victim visits a webpage that contains a hidden <iframe> which redirects the victim's browser to another redirection page or to the HTTP server hosting the exploit pack. The exploit pack attempts to exploit a list of vulnerabilities on the victim computer. If exploitation is achieved then the victim computer maintains a "keep alive" state with the exploit pack server. The exploit pack has opened a door and is keeping it open. At this point a malicious payload may be pushed to the victim for installation. These payloads typically involve a bot binary. Once the binary is installed, the victim machine joins the intended HTTP botnet. There are many different variations on this drive-by exploitation life cycle, but the aforementioned steps are typical.

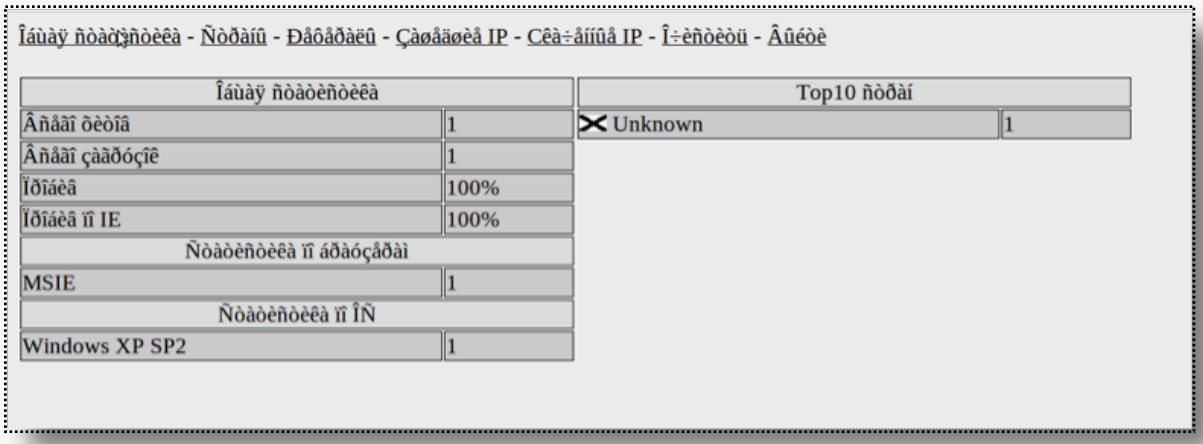


iframe infection graph

In evaluating the potency and effectiveness of any particular exploit pack, we recognized that our testing environment was incomplete without using the specified HTTP referrers that might prevent the exploit pack from executing the correct routine. The newer exploit packs will only perform if the HTTP referrer is correct, the assumption being that without the correct referrer (the designed HTTP redirector) the index page visit is hostile – a wayward web visit at best and a security researcher at worst.

Gpack and Adpack infected our client via Internet Explorer and subsequently registered the infection statistic in the control panel even though we were unable to read the actual text.

G-Pack text t unreadable

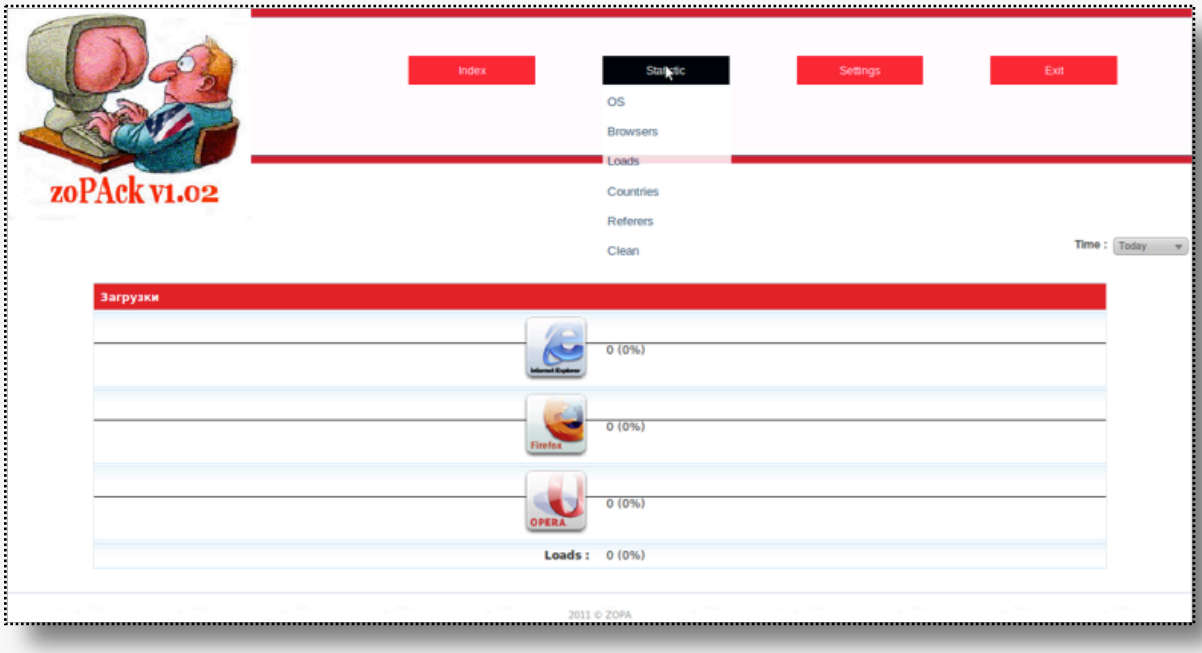


Adpack unreadable text

Non-English language in Siberia pack

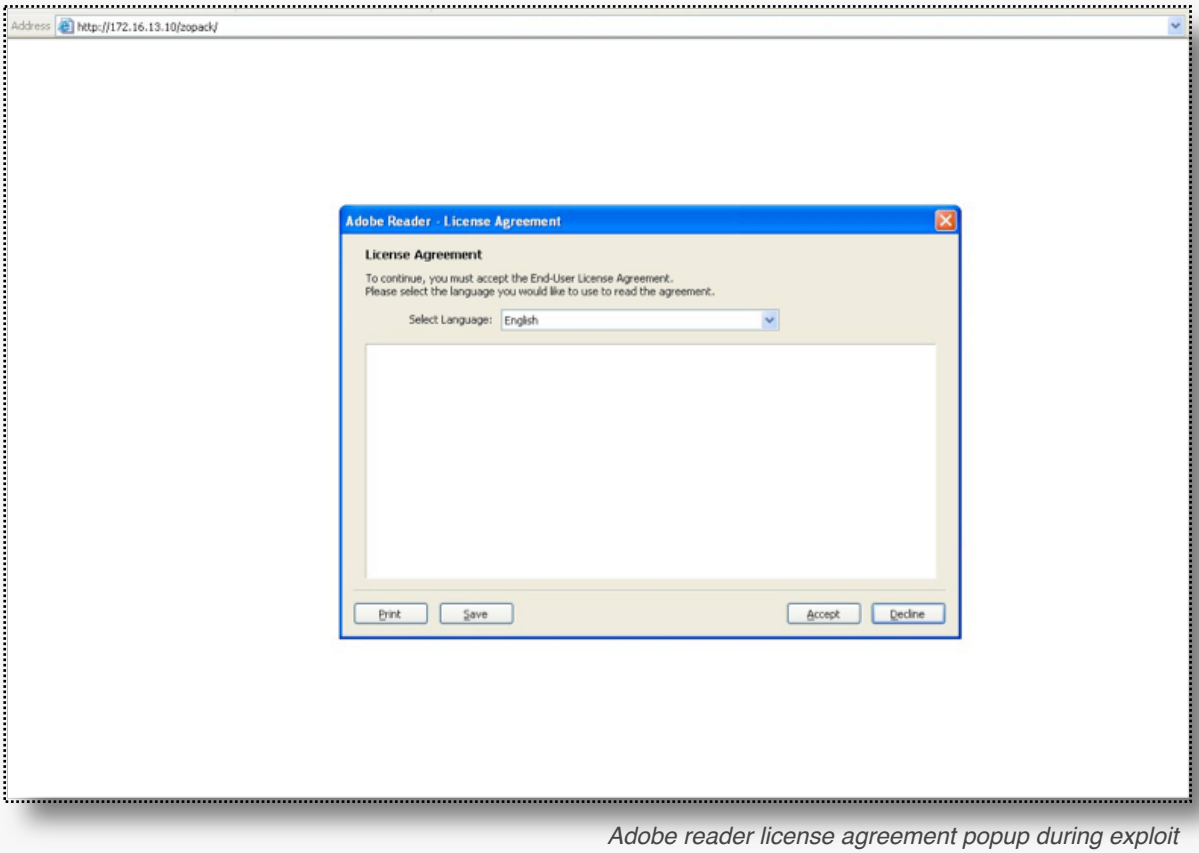
A number of exploit packs compromised our victim client though the database exploitation statistic never registered. Zopack (anti-American sentiment included) was one such example

ZoPack Logo



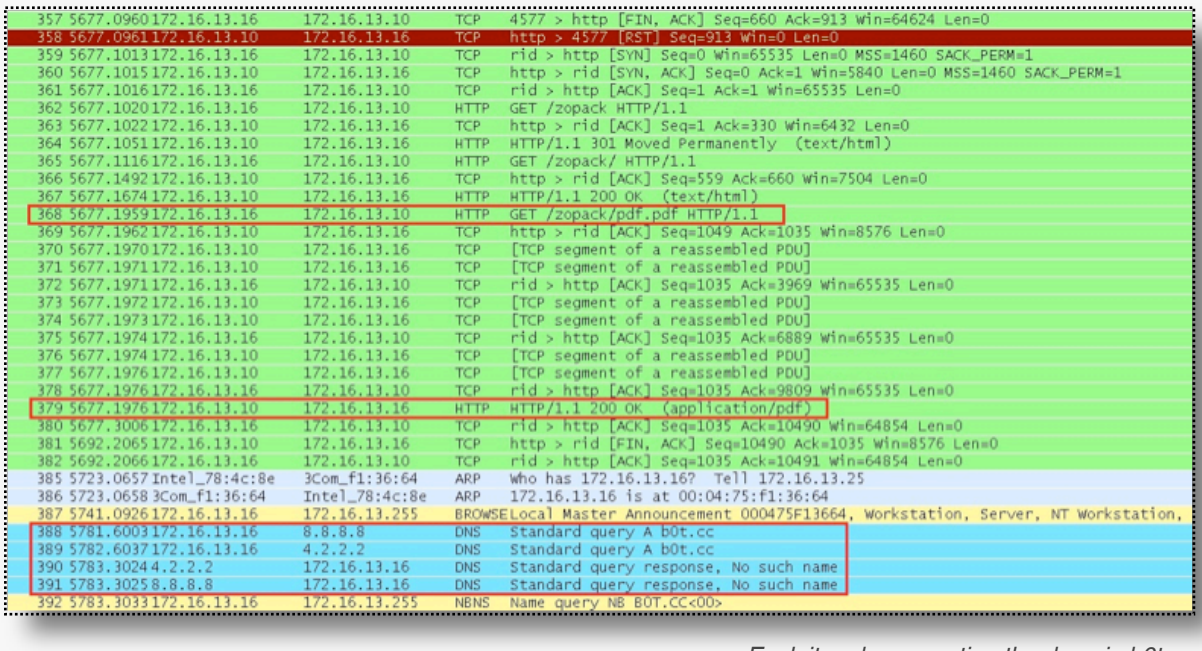
ZoPack statistics page

After visiting the Zopack index.php page the only visual indication of exploit activity was the Adobe Reader license agreement dialogue box which was only present because the application had never been opened (as set during image creation). Otherwise, from a victim perspective the exploitation was completely quiet.



Adobe reader license agreement popup during exploit

Immediately after the pdf.pdf file was transferred and our machine was exploited, the client made a DNS A record query for b0t.cc which was no longer a valid domain at the time of analysis.



357	5677.0960	172.16.13.16	172.16.13.10	TCP	4577 > http [FIN, ACK] Seq=660 Ack=913 Win=64624 Len=0
358	5677.0961	172.16.13.10	172.16.13.16	TCP	http > 4577 [RST] Seq=913 Win=0 Len=0
359	5677.1013	172.16.13.16	172.16.13.10	TCP	rid > http [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
360	5677.1015	172.16.13.10	172.16.13.16	TCP	http > rid [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 SACK_PERM=1
361	5677.1016	172.16.13.16	172.16.13.10	TCP	rid > http [ACK] Seq=1 Ack=1 Win=65535 Len=0
362	5677.1020	172.16.13.16	172.16.13.10	HTTP	GET /zopack HTTP/1.1
363	5677.1022	172.16.13.10	172.16.13.16	TCP	http > rid [ACK] Seq=1 Ack=330 Win=6432 Len=0
364	5677.1051	172.16.13.10	172.16.13.16	HTTP	HTTP/1.1 301 Moved Permanently (text/html)
365	5677.1116	172.16.13.16	172.16.13.10	HTTP	GET /zopack/ HTTP/1.1
366	5677.1492	172.16.13.10	172.16.13.16	TCP	http > rid [ACK] Seq=559 Ack=660 Win=7504 Len=0
367	5677.1674	172.16.13.10	172.16.13.16	HTTP	HTTP/1.1 200 OK (text/html)
368	5677.1959	172.16.13.16	172.16.13.10	HTTP	GET /zopack/pdf.pdf HTTP/1.1
369	5677.1962	172.16.13.10	172.16.13.16	TCP	http > rid [ACK] Seq=1049 Ack=1035 Win=8576 Len=0
370	5677.1970	172.16.13.10	172.16.13.16	TCP	[TCP segment of a reassembled PDU]
371	5677.1971	172.16.13.10	172.16.13.16	TCP	[TCP segment of a reassembled PDU]
372	5677.1971	172.16.13.16	172.16.13.10	TCP	rid > http [ACK] Seq=1035 Ack=3969 Win=65535 Len=0
373	5677.1972	172.16.13.10	172.16.13.16	TCP	[TCP segment of a reassembled PDU]
374	5677.1973	172.16.13.10	172.16.13.16	TCP	[TCP segment of a reassembled PDU]
375	5677.1974	172.16.13.16	172.16.13.10	TCP	rid > http [ACK] Seq=1035 Ack=6889 Win=65535 Len=0
376	5677.1974	172.16.13.10	172.16.13.16	TCP	[TCP segment of a reassembled PDU]
377	5677.1976	172.16.13.10	172.16.13.16	TCP	[TCP segment of a reassembled PDU]
378	5677.1976	172.16.13.16	172.16.13.10	TCP	rid > http [ACK] Seq=1035 Ack=9809 Win=65535 Len=0
379	5677.1976	172.16.13.10	172.16.13.16	HTTP	HTTP/1.1 200 OK (application/pdf)
380	5677.3006	172.16.13.16	172.16.13.10	TCP	rid > http [ACK] Seq=1035 Ack=10490 Win=64854 Len=0
381	5692.2065	172.16.13.10	172.16.13.16	TCP	http > rid [FIN, ACK] Seq=10490 Ack=1035 Win=8576 Len=0
382	5692.2066	172.16.13.10	172.16.13.16	TCP	rid > http [ACK] Seq=1035 Ack=10491 Win=64854 Len=0
385	5723.0657	Intel_78:4c:8e	3Com_f1:36:64	ARP	Who has 172.16.13.16? Tell 172.16.13.25
386	5723.0658	3Com_f1:36:64	Intel_78:4c:8e	ARP	172.16.13.16 is at 00:04:75:f1:36:64
387	5741.0926	172.16.13.16	172.16.13.255	BROWSE	Local Master Announcement 000475F13664, Workstation, Server, NT Workstation,
388	5781.6003	172.16.13.16	8.8.8.8	DNS	Standard query A b0t.cc
389	5782.6037	172.16.13.16	4.2.2.2	DNS	Standard query A b0t.cc
390	5783.3024	4.2.2.2	172.16.13.16	DNS	Standard query response, No such name
391	5783.3025	8.8.8.8	172.16.13.16	DNS	Standard query response, No such name
392	5783.3033	172.16.13.16	172.16.13.255	NBNS	Name query NB B0T.CC<00>

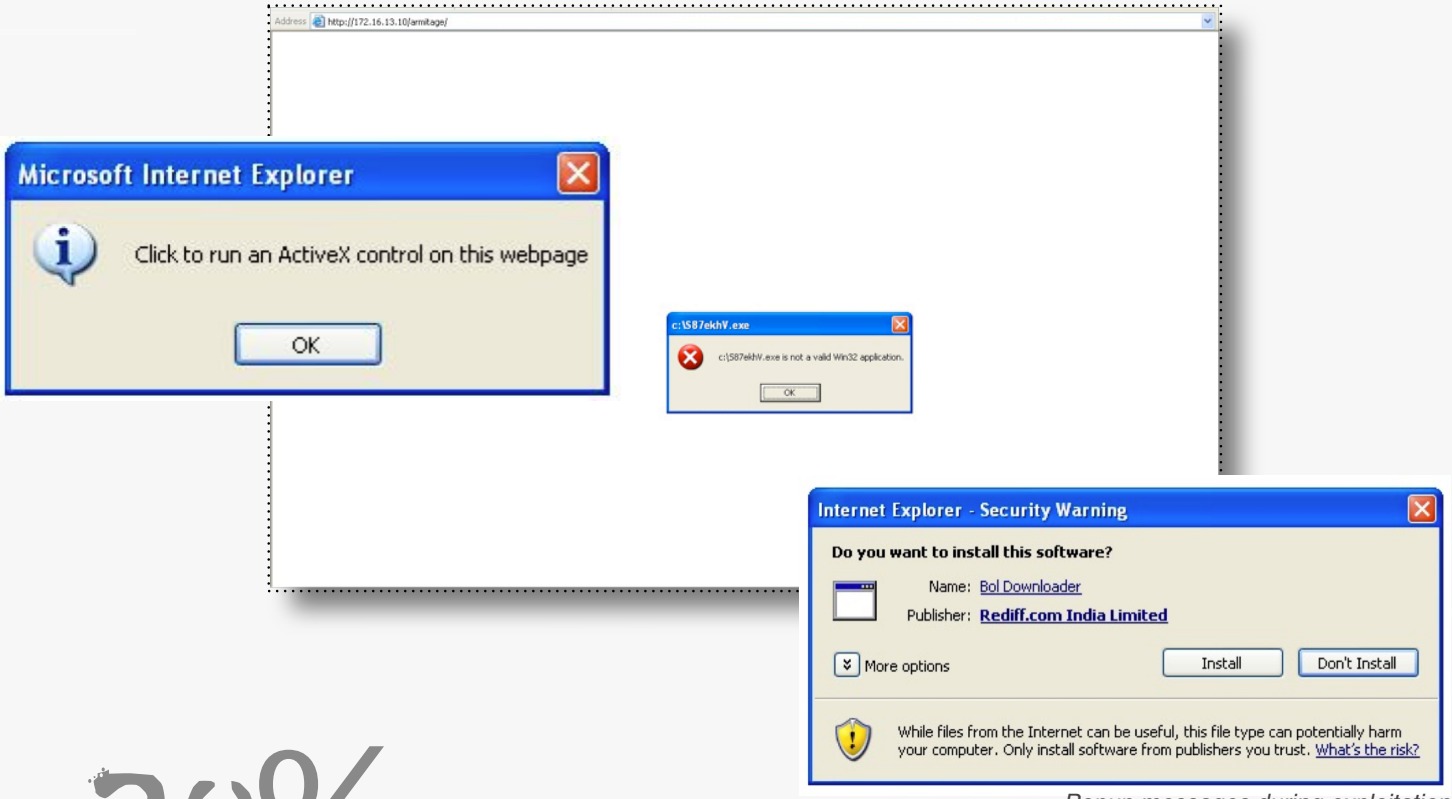
Exploit code requesting the domain b0t.cc

Sploit25 was another exploit pack that infected our client, and did not increment the control panel infection counter. The only visual sign of exploitation was Adobe Reader momentarily embedded in Internet Explorer.

Some exploit pack HTTP landing pages were blank during exploitation and others displayed a 404 message like Gpack. A blank page containing an embedded iframe with a 1x1 pixel was invisible and therefore unnoticed by our victim.

Sploit25 showing a blank page

Other exploit packs including Armitage, Eleanore, and Icepack all displayed various dialogue boxes during exploitation.



Popup messages during exploitation

30%

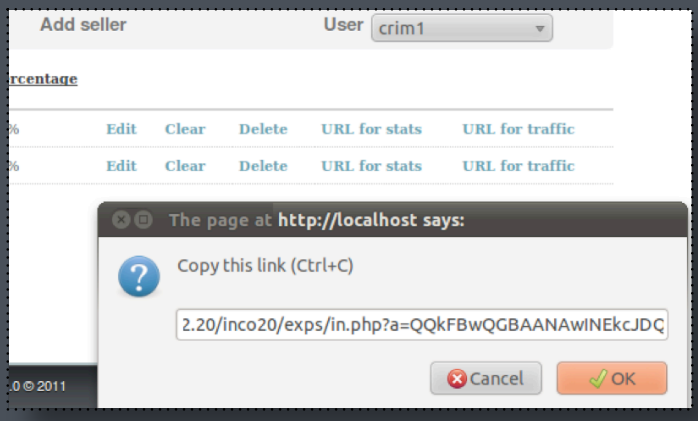
Overall, we encountered a 30% infection rate "out of the box". A number of variables were likely responsible for an exploit pack's failure to exploit our victim computer. The two most prevalent reasons were missing files and/or incorrect HTTP referrers.

MONETIZATION

A growing number of exploit packs have incorporated functionality that adds additional users or sellers to the database. Fragus was one of the first exploit packs that allowed the administrator to add a seller and provide him with a unique URI. The control panel interface allowed the administrator to track installations originating from any particular seller[8]. A seller could drive traffic to the exploit pack administrator and receive the associated compensation.

Fragus exploit pack add seller option

The URI provided to the seller was comprised of a variable within a link. The link variables we observed were typically hash values. These hash values were based on user specific settings or a randomly generated string. This allowed the administrator to track and pay for the number of installations by individual seller.



The applicable exploit packs also provided sellers with a personalized statistics page which displayed installations and HTTP visits.

[8] Some exploit packs referred to these individuals as “resellers”.

Incognito (and other recent packs) offered an option to rent the pack to multiple users, which allowed the exploit pack owner to act as a broker. The system tracked the 'users' and connected 'sellers' by supplying separate statistics for each seller and login function for each user as well as provided options to host different executables for each user.

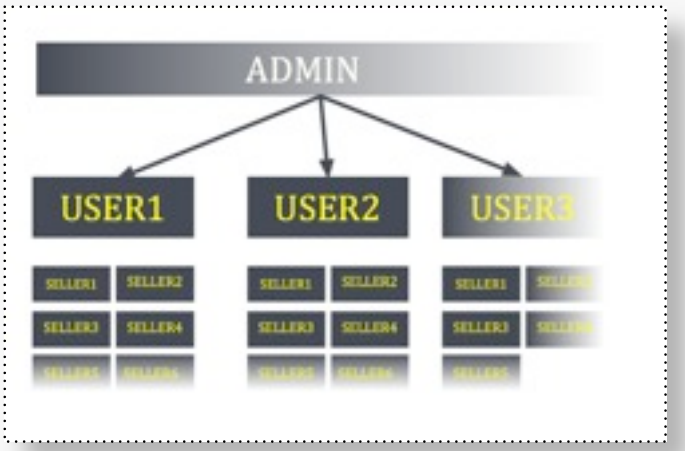
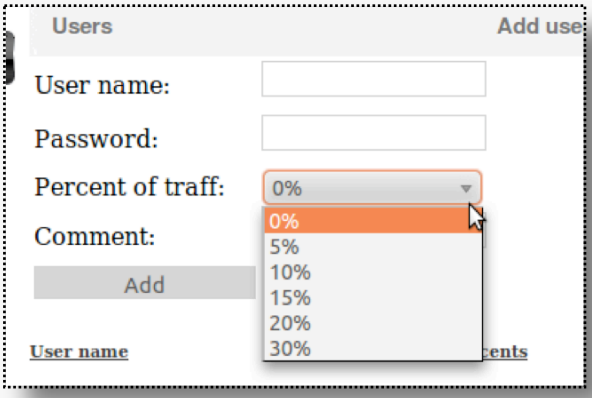


Chart showing user and seller layout in Incognito

Since the creation of Pay-Per-Install (PPI), "shaving", - a percentage of installations are subtracted from the actual total – has been a favorite criminal pastime. This practice regularly leads to conflicts between the 'seller' and the 'controller' as statistics fail to properly coalesce. We have observed botnets were 90% of the traffic was 'shaved' from the bot herder's counter.

During Incognito analysis we noticed that the administrator interface contained an option to set a 'shave' percentage for each user.



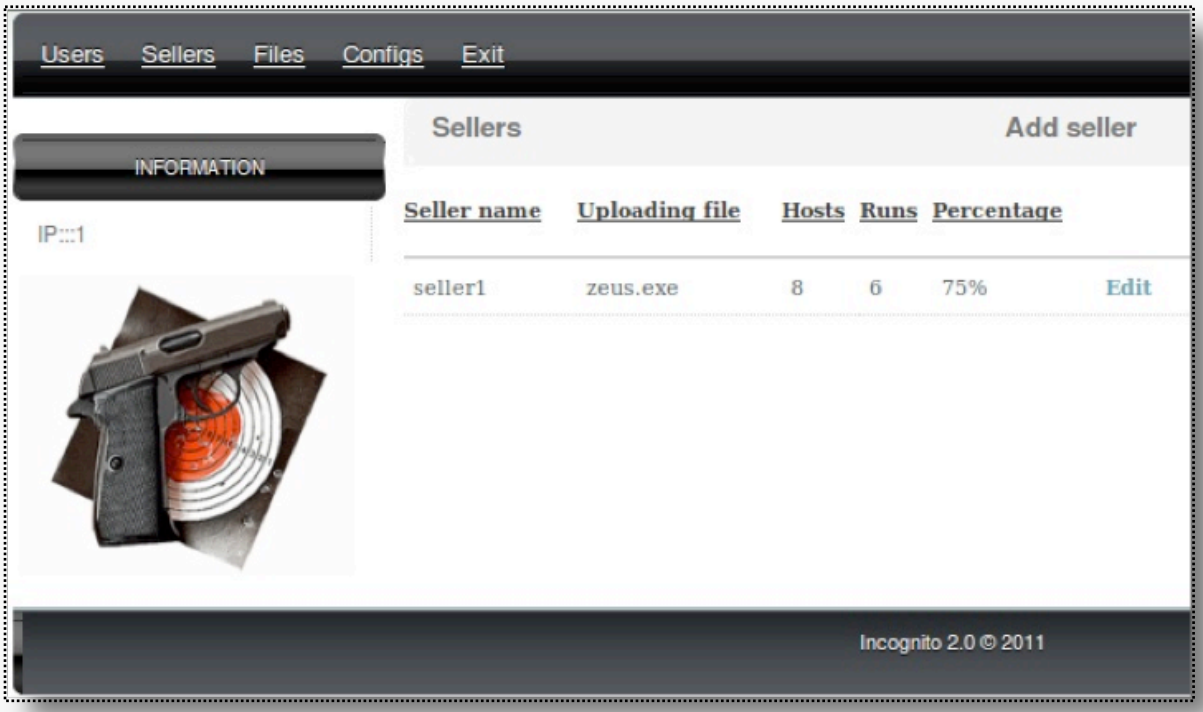
Shave percentage settting in Incognito

The first infection point within the exploit code page was a 'percent of traff' value which was used to discount HTTP visit counters from the sellers' user group. In our particular instance, 30% of the total users' traffic was unrecognized.

```
elseif ($percent == '4' && mt_rand(1,5) == 1)
{
        $UserInfo['u'] = 0;
        $UserInfo['s'] = $UserTraffId['u'];
}
elseif ($percent == '5' && mt_rand(1,3) == 1)
{
        $UserInfo['u'] = 0;
        $UserInfo['s'] = $UserTraffId['u'];
```

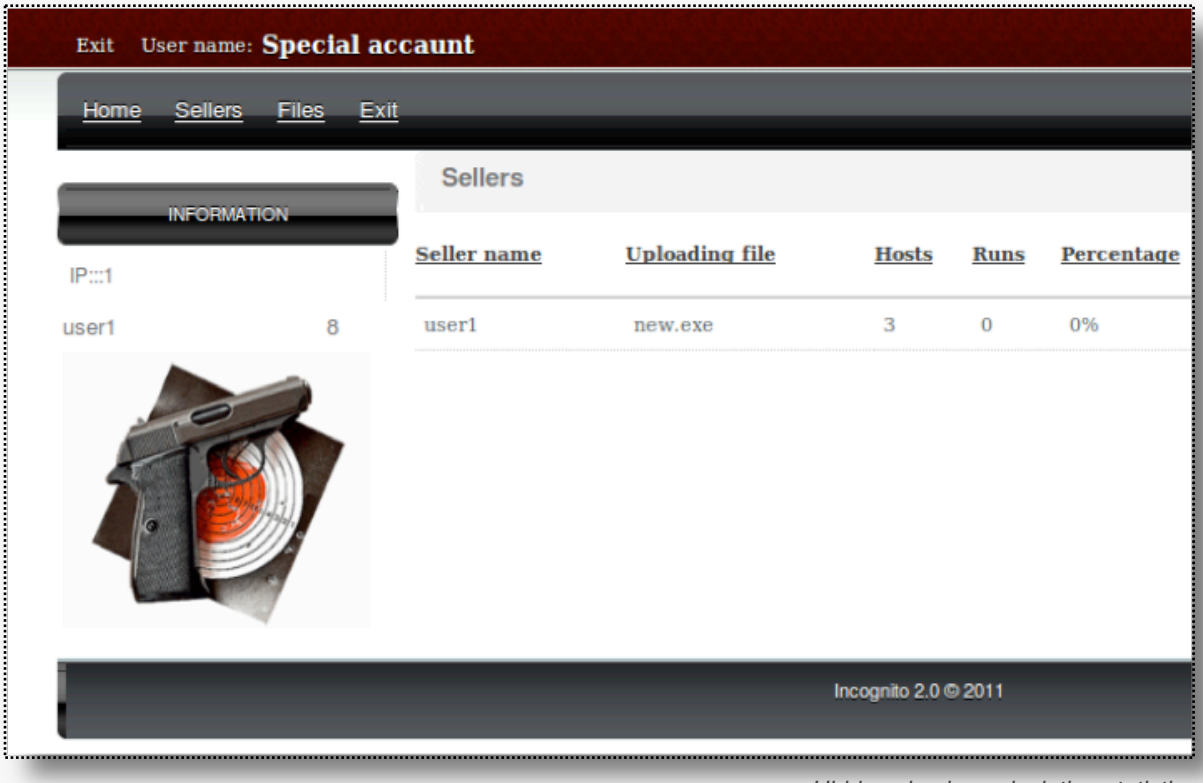
Landing page PHP code showing 'shaving' calculation

These nulled hits' were viewable on a designated page. This 'Special Accaunt' titled page allowed the administrator to set a special executable for each user in the system. The administrator was siphoning 'nulled' traffic from system users for additional profits.



Statistics page for normal user

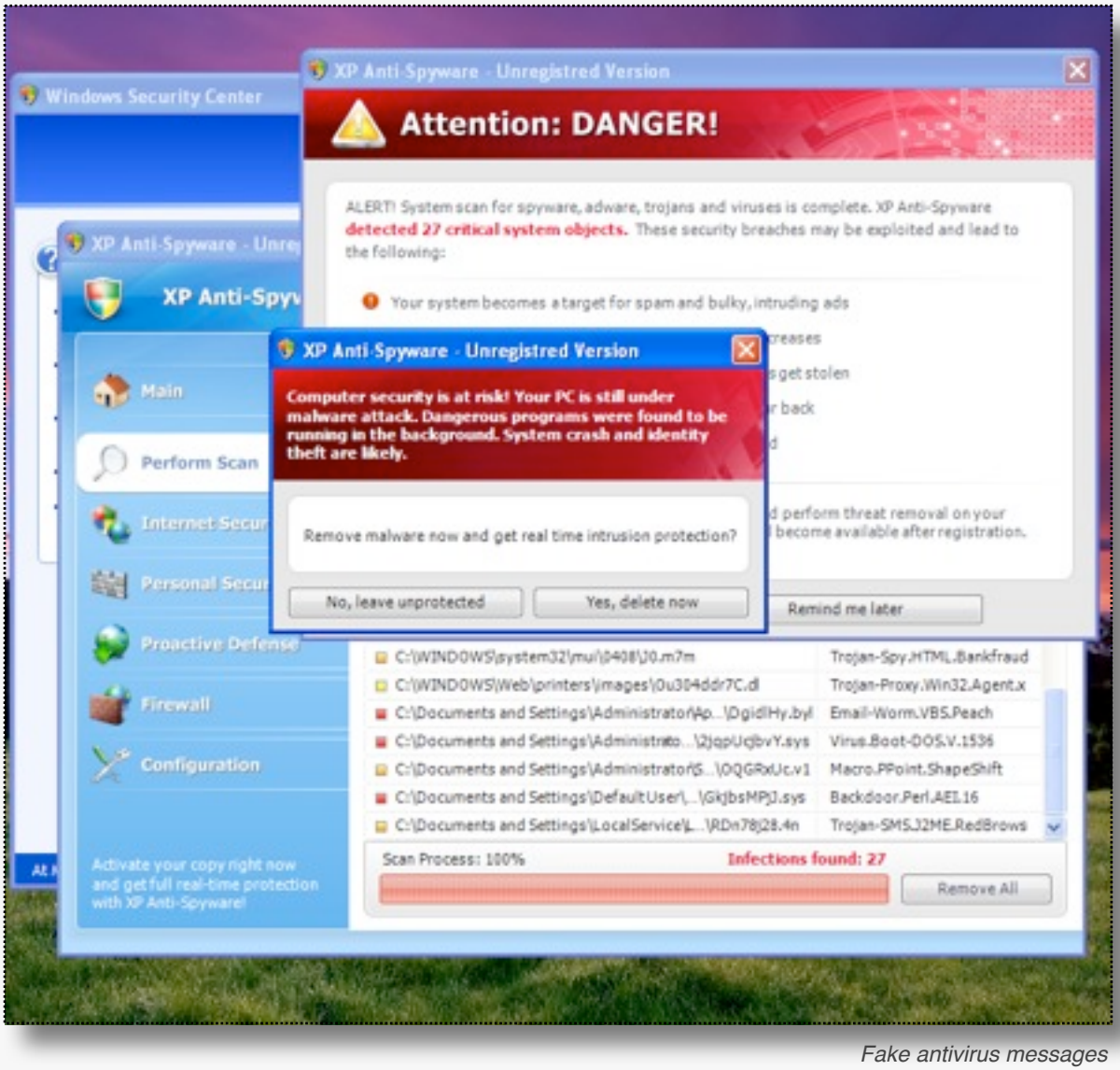
We successfully infected 6 machines (see above) out of 8 total hosts that visited our exploit page. If our Incognito instance was live on the Internet these statistics are what the seller would have seen. However the 'special accaunt' page displayed the actual traffic statistics for 'user1' as 11 (8+3). These 3 visits represented the shaved traffic percentage that was available for further administrator redirection.



Hidden shaving calculation statistics

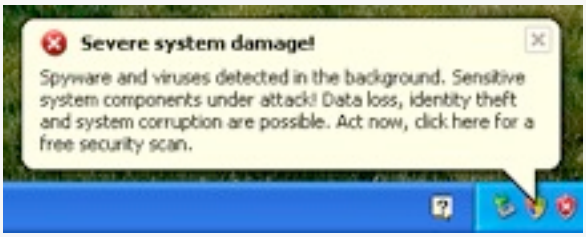
So how are these 'runs' monetized? The exploit pack's objective is to install an executable on the exploited client. These executables often fall in the banking Trojan class like SpyEye or Carberp, which intercept online banking transactions.

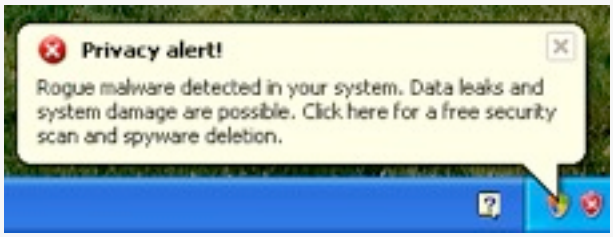
Fake antivirus scareware is another popular executable pushed via exploit packs. Below is a fake antivirus package that was installed subsequent to our machine's infection via a Black hole exploit pack. The scareware caused a constant stream of system infection messages while at the same time the browser was rendered useless with virus notifications.



Fake antivirus messages

This fake anti-virus installation would count as a 'run' for the 'seller' and translate into a micro-payment, which when multiplied by thousands of victim computers equals real money.





CODE PROTECTION

An exploit pack is valued by its effective exploitation rate, support availability, and value added functionality. When exploit pack code is leaked to the criminal public, the user value immediately diminishes. Exploit pack authors all have the same problem: the buyer has access to the source code. There is also the possibility that a buyer might marginally modify the code and sell it as a different competing family/brand. To protect their intellectual property, some exploit pack writers have relied on PHP source code encoding.

The Blackhole exploit pack is one example where the source was protected using Ioncube.[9] Zend is a different encoding product used by other exploit packs. When Ioncube is used, the technical skills required to setup a working exploit pack slightly increase as the Ioncube libraries must be correctly installed. A default LAMP installation is not sufficient to host the code.

These tools do provide a layer of protection and criminals are constantly searching for methods to defeat the protection. This has created demand for a 'dezending' or 'decoding' service. In our experience these services produced mixed results. While we observed successful 'dezending' examples, Ioncube decoding appeared to be more challenging. Several freeware tools have appeared in the public domain advertising Ioncube decoding capabilities, but our analysis indicated that one of those tools was only partially able to decrypt the code, leaving the majority of it obfuscated and unusable.

Tools like Ioncube offer the exploit pack author or re-seller additional advantages by restricting the code's usability to a specific IP address or domain name. For each visit, the Ioncube code check's if the page is hosted on a server using the correct domain or IP address. It can also perform a sanity check based on the server's system time. This presents the exploit pack market with a new type of commodity. The code can be conditionally sold and prices set accordingly. For example, underground advertisements tout exploit packs for sale with a 6 month license that is limited to 1 specific domain.

[9] hxxp://www.ioncube.com - Ioncube is a completely legitimate tool which offers programmers PHP code / intellectual property protection.

FUTURE

In 2005, the authors of browser exploit packs had little patience for users who were incapable of using Linux, setting up a MySQL database, or correctly securing an Apache installation. Six years later BEP authors want to sell to non-technical fraudsters. These users are not primarily interested in the theft and replication of the pack and they are happy to rent an exploit pack installation and forgo the configuration headaches.

Criminal SaaS (Software as a Service) is not a new concept, and exploit pack authors may look to create new packs using server side frameworks like Rails where buyers can host on managed services such as bullet proof hosting criminal versions similar to Heroku. These systems allow for a quick and scalable deployment while at the same time keeping the back end code opaque.

We see no reason for the exploit pack's (especially managed) popularity to decrease in coming years. Users who patch and update are generally immune to drive by attacks, unless a previously unknown exploit is in use. Given the security community's quick response to any new threat, even a pack containing a "0 day" exploit probably loses its initial potency within 48 hours.

Future packs will continue to bundle stale and new exploits while improving the control panel interface and adding easy use features that enable fraudsters to diversify their revenue streams. We believe exploit pack authors will delay development work on mobile browser exploit packs until criminal demand for traditional packs decreases significantly across the Underground.

Authors: Levi Gundert & Marcel van den Berg
Team Cymru Business Intelligence Team, 16W361 S. Frontage Road Suite 100, Burr Ridge, IL 60527 US