Know Your Enemy: Behind the Scenes of Malicious Web Servers

The Honeynet Project
http://www.honeynet.org

Christian Seifert – *The New Zealand Honeynet Project*

Last Modified: *7 November 2007*

INTRODUCTION

In our recent KYE paper on malicious web servers, we identified several hundred malicious web servers. These servers launched, so-called drive by downloads, that allowed them to gain complete control of the client machine without the consent or notice of the user, who merely visited the malicious web server with his (vulnerable) web browser. In our study, we analyzed a large number of web servers with our client honeypot Capture-HPC, which allowed us to assess whether a server was malicious, then inspect the exploit code that was sent to the client and the potential malware downloaded. However, many questions remained unanswered:

1. We observed that malicious servers were not consistent in their behavior. When interacting with a malicious server, it might initially demonstrate malicious behavior, but cease to do so on subsequent attempts. We were unable to discover with certainty the reason or technique for this non-deterministic behavior.
2. We observed that only the Internet Explorer browser was targeted. Was this because attackers were choosing not to attack the other browsers in our study, or because of the specific set of browsers/versions we chose to include?
3. We observed that malicious web pages accessed centralized exploit servers. However, we were unable to determine whether this was common practice or a one-time incident.
4. We consistently observed obfuscation to be deployed on the malicious pages, but could determine neither how the obfuscated code was generated nor whether there are elements of the obfuscation engine that are consistent and detectable with static code analysis.

Web exploitation kits, which increasingly appeared in 2006/7, will provide us with a behind-the-scenes look at how these malicious web servers operate. In this paper we will give a brief functional overview of several web exploitation kits, then dwelve into answering the questions above through analysis of these kits and malicious web servers that use it. The web exploitation kits that we will examine are Webattacker, MPack and Icepack. We conclude with implications of our discoveries on client honeypot technology and future studies on malicious web servers.

WEB EXPLOITATION KITS OVERVIEW

A web exploitation kit allows an attacker to gain control of a client machine when it visits a malicious web page. Figures 1 and 2 show the steps that are usually taken by these drive-by-downloads. First, a user visits a web page that hosts a web exploitation kit. Following the client's request, the web server might implement some server side logic that assesses from what country the request is coming, what browser is being used, etc. and then returns the attack code as part of the response. The attack code attacks the client, and if successful, executes a downloader component without the user's consent or notice (Step 1). The downloader in turn will make a follow-on request to download and execute a piece of malware from a URL specified by the attacker (Step 2). Alternatively, the step can be skipped and the malware delivered with the initial attack code making Step 2 unnecessary. At that point, the attacker has complete control of the client machine and can steal sensitive information, such as credit card numbers or account credentials, join the client machine in a botnet, use social engineering to entice the user to purchase bogus products online, etc. The user did not notice that he was just successfully attacked as all steps are happening in the background. As such, drive-by-downloads is a popular technique in the attacker's arsenal.

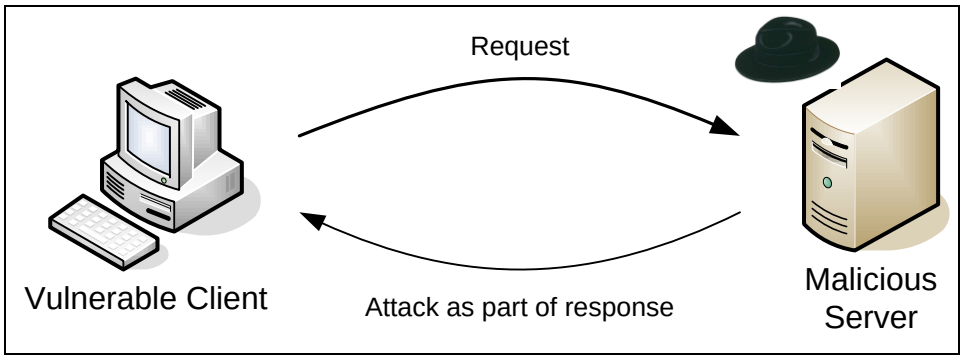

Figure 1 – Client-Side Attack - Step 1

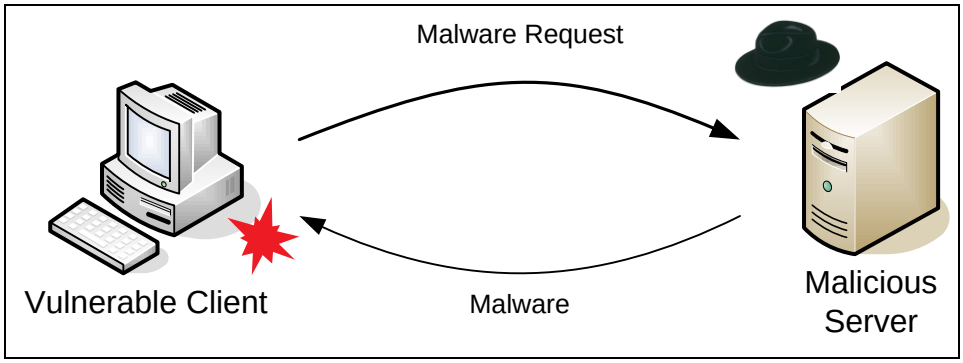

Figure 2 – Client-Side Attack - Step 2

WebAttacker is one of the first web exploitation kits that appeared in early 2006. It was sold on Russian web sites for about $15 [1]. MPack, a more sophisticated web exploitation kit, was developed shortly after by three Russian programmers who call themselves the "Dream Coders Team" [2], and was initially released in June 2006. The tool is sold via the underground market for approximately $700 to $1000 and according to VeriSign/iDefense [3], it has been responsible for thousands of infections. IcePack was first reported in July 2007 [4,5] and is quite similar to MPack. It has been developed by another group, the IDT group, with a purchase price of $400. Since August 2007 even Chinese localizations of the MPack/IcePack toolkits are available [6].

Under the hood, these web exploitation kits are quite simple. WebAttacker consists of a Perl and some PHP scripts. MPack/IcePack consists of several PHP scripts and a downloader creator that allows the user to create custom downloaders, programs designed to retrieve and install the actual malware. The use of the downloader frees the attacker from any size limits posed by the payload buffer and potentially provides some encryption routines to evade intrusion detection systems. Our test installation of MPack was as easy as unpacking the MPack archive into a directory, editing a simple configuration file and placing the downloader into the directory. The provided documentation and sample configuration assisted us in our efforts. Once the web exploitation kit is installed, attacks are live and accessible on the web server under a specific URL, and the attacker's only remaining task is to entice users to visit this URL.

Once a kit is set up, the application can provide the attacker with information about the progress of its attacks via a password protected administrative/statistics page. The MPack administrative interface is shown in Figure 3. It primarily contains information on the success rate of the various attacks and information about the location of the attacked clients. Similar administrative interfaces exist for WebAttacker and IcePack. We will review the administrative interface of MPack in more detail below.

IP TRACKING

First, we will investigate our observation of non-deterministic behavior of malicious web servers. Repeated interaction with a malicious web server did not consistently yield malicious behavior. Analysis of MPack/ IcePack exploitation kits allows us to - at least partially - explain this behavior.

[1] http://www.theregister.co.uk/2006/03/27/spyware_diy/
[2] http://www.theregister.co.uk/2007/07/23/mpack_developer_interview/
[3] http://isc.sans.org/diary.html?storyid=3015
[4] http://www.finjan.com/MCRCblog.aspx?EntryId=1601
[5] http://pandalabs.pandasecurity.com/archive/Ice_2800_Pack_2900_-for-the-summer.aspx
[6] http://ddanchev.blogspot.com/2007/10/mpack-and-icepack-localized-to-chinese.html

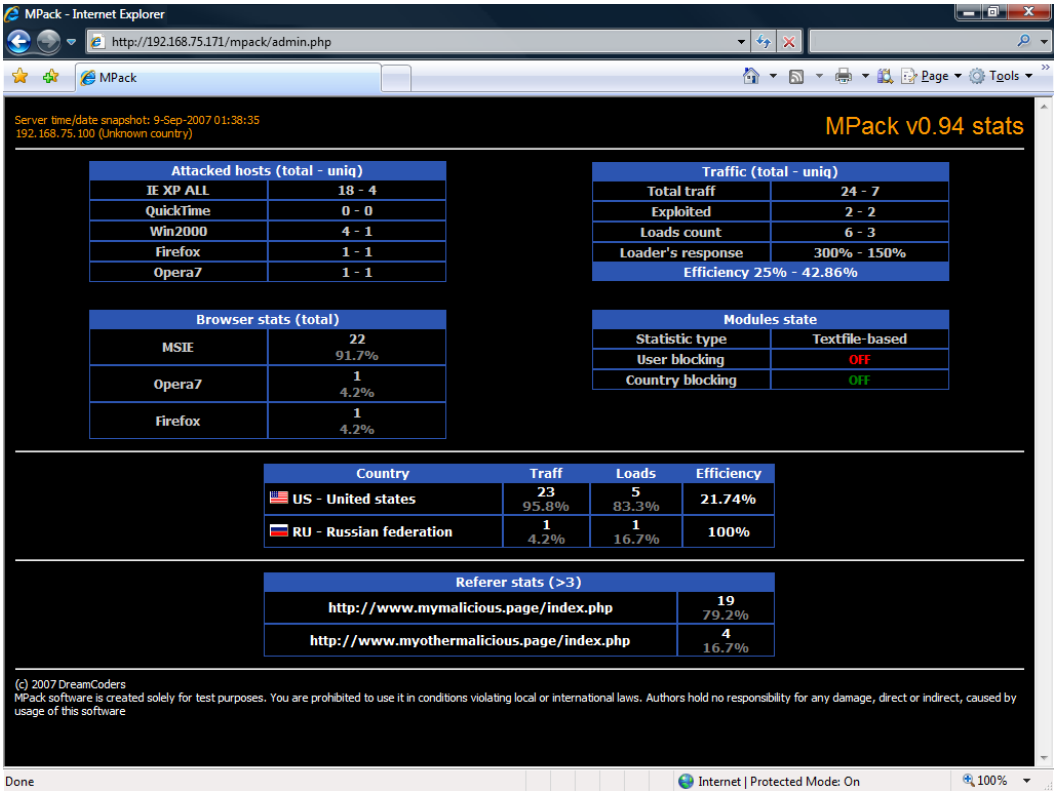

Figure 3 - MPack Administrative Interface

MPack can be configured via the `$BlockDuplicates` option to only deliver an attack to a user it hasn't seen before. If this "IP tracking functionality" is enabled, MPack exectutes the `CheckAddUser` function shown in Figure 4. The user's IP address with the browser identifier are hashed (line 08) and stored in the MPack database (mysql or txt file) (lines 48-49 and 23-25). Upon repeated visits by the user with the same IP address and browser identifier, the hash is once again generated and checked for existence in the MPack database (lines 15 and 29, 30, 41-42). If it is found, an "unhappy" emoticon is displayed and no attack is delivered (line 17 and 44). (Similar functionality exist in IcePack [7].)

The CheckAddUser function explains why certain URLs are malicious and then permanently go dormant. However, this would not explain URLs that exhibit malicious behavior, temporarily go dormant (maybe one or two requests), but then exhibit malicious behavior once again; this is a pattern we observed in our study. To explain this observation, we turn to an additional attack technique of Fast-Flux networks, which we more extensively describe in our Know Your Enemy: Fast-Flux Service Networks paper. Fast-Flux networks are networks computer systems with public DNS records that are constantly changing. If a malicious URL is part of such a network, it might resolve to actual different physical machines, which will only have access to their local IP tracking database. If a client honeypot accesses the same URL repeatedly, it might actually interact with several physical machines that each trigger initially, but then permanently go dormant. From the client honeypot's view, however, it appears as if it is sporadically attacked.

[7] http://www.finjan.com/MCRCblog.aspx?EntryId=1601

```
00:
01: //checks and saves user's IP hashed with browser
02: //to avoid future browser's hangup
03: function CheckAddUser()
04: {
05: global $UseMySQL;
06: global $dbstats;
07:
08: $ipua=md5(getenv("REMOTE ADDR").getenv("HTTP USER AGENT"));
09:
10: if ($UseMySQL==0) {
11: //text variant
12:     $fn="users.txt";
13:     if (file_exists($fn)) {
14:     $lines = file($fn);
15:             if (in array($ipua."\n", $lines)==TRUE) {
16:             //got dup
17:             echo ";[";
18:             exit;
19:             }
20:     }
21:
22:     //uniq record
23:     $fp=fopen($fn,"a");
24:     fwrite($fp,$ipua."\n");
25:     fclose($fp);
26: } else {
27:
28: //mysql variant
29:  $query = "SELECT * FROM ".$dbstats."_users WHERE data='".$ipua."'";
30:  $res=mysql_query($query);
31:  $merr=mysql error();
32:     if ($merr!="") {
33:             //looks like no table, create & add data
34:             $query="CREATE TABLE `".$dbstats."_users` (`data` VARCHAR( 32 ) NOT NULL ) ENGINE =
                        MYISAM ;";
35:             mysql_query($query);
36:               $query = "INSERT INTO ".$dbstats." users VALUES ('".$ipua."')";
37:               mysql_query($query);
38:
39:     } else {
40:     //table found, check returned set count
41:     $rcount=@mysql_num_rows($res);
42:     if ($rcount>0) {
43:             //found data, prevent view
44:             echo ":[";
45:             exit;
46:     } else {
47:             //not found, add
48:             $query = "INSERT INTO ".$dbstats." users VALUES ('".$ipua."')";
49:             mysql query($query);
50:     }
51:     }
52:
53: }
54:
55: }
```

Figure 4 - CheckAddUser Function

TARGETS

In our study on malicious web servers, we interacted with web servers using older versions of the browsers Mozilla Firefox, Opera and Microsoft Internet Explorer. We didn't configure any plug-ins other than the ones provided by a clean Windows XP SP2 installation. The results of our study yielded only observed attacks on Internet Explorer, but did not observe any attacks on Firefox and/or Opera. However, just because we didn't observe any attacks on these clients, we couldn't conclude whether they are not targeted at all, because our configuration/versions might have simply been too bare or old.

The web exploitation kits provides us with a specific answer to this question. Other browsers and operating systems are certainly targeted as well. The following attacks are currently supported by the web exploitation kits. They first assess which browser and operating system are being used before delivering an attack to foster a high success rate:

Target	WebAttacker (as of September 2006)	MPack V0.94	IcePack[8] (as of September 2007)
IE	Microsoft Data Access Component Vulnerability (CVE-2006-0003) Windows VML Vulnerability (CVE-2006-4868) Microsoft Virtual Machine Vulnerability (CVE-2003-0111)	Microsoft Data Access Component Vulnerability (CVE-2006-0003) Apple QuickTime RTSP URI Remote Buffer Overflow Vulnerability (CVE-2007-0015) WinZip FileView ActiveX Control Multiple Vulnerabilities (CVE-2006-6884) Microsoft WebViewFolderIcon ActiveX Control Buffer Overflow Vulnerability (CVE-2006-3730) Microsoft Management Console Vulnerability (CVE-2006-3643)	Microsoft Data Access Component Vulnerability (CVE-2006-0003) WebViewFolderIcon ActiveX Control Buffer Overflow Vulnerability (CVE-2006-3730) Microsoft Management Console Vulnerability (CVE-2006-3643) Vector Markup Language Vulnerability (CVE-2007-0024) Microsoft DirectX Media 6.0 Live Picture Corporation DirectTransform FlashPix ActiveX (CVE-2007-4336) Yahoo! Messenger Webcam ActiveX Remote Buffer Overflow Vulnerability (CVE-2007-3147, CVE-2007-3148) Yahoo! Widgets YDP ActiveX Control Buffer Overflow Vulnerability (CVE-2007-4034)

[8] http://blog.trendmicro.com/icepack-packing-heat/

Target	WebAttacker (as of September 2006)	MPack V0.94	IcePack[9] (as of September 2007)
Opera	Windows Media Player Plug-In with Non-Microsoft Internet Explorer Vulnerability (CVE-2006-0005)	Windows Media Player Plug-In with Non-Microsoft Internet Explorer Vulnerability (CVE-2006-0005)	Windows Media Player Plug-In with Non-Microsoft Internet Explorer Vulnerability (CVE-2006-0005)
Firefox	Exploitable crash in InstallVersion.compareTo vulnerability (CVE-2005-2265) Windows Media Player Plug-In with Non-Microsoft Internet Explorer Vulnerability (CVE-2006-0005)	Windows Media Player Plug-In with Non-Microsoft Internet Explorer Vulnerability (CVE-2006-0005)	Windows Media Player Plug-In with Non-Microsoft Internet Explorer Vulnerability (CVE-2006-0005) JavaScript Navigator Object Vulnerability (CVE-2006-3677)

Table 1 - Supported Attacks

While the set of attacks supported is rather small, it does show that multiple browsers are targeted. Many attacks utilize attack vectors that make use of plug-ins. Firefox and Opera are being attacked via a Microsoft Windows Media Player plug-in and Internet Explorer is attacked via some older vulnerabilities of the browser (MDAC attack), but also some more recent vulnerabilities in browser plug-ins, such as QuickTime and Winzip [10]. In September 2007, a vulnerability was discovered that allows an attacker to execute arbitrary code on Internet Explorer 7 via crafted PDF files (CVE-2007-5020); another dangerous potential candidate for inclusion in one of the kits. How quickly new releases with new attacks are actually unleashed is still not well understood and will be answered as part of future work.

If we look at these three web exploitation kits as evolutionary successions, however, there seems to be a trend towards attacking plug-ins. Browsers nowadays have some sort of automated update mechanism that would result in a closure of the attack window for these web exploitation kits fairly quickly. However, third-party plug-ins, such as Flash or Yahoo! Widgets, do not have such update mechanisms and remain unpatched on the system even when patches are available. Targeting these plug-ins ensures that the web exploitation kits remain effective.

[9] http://blog.trendmicro.com/icepack-packing-heat/
[10] Note that this the attack sequence we encountered on the Italian Keith Jarrett web site.

MPack has an additional feature that we haven't considered so far: geolocation-dependent triggering via the freely available MaxMind Geolocation Technology. MPack determines in what country the user is located and can be configured to only trigger on certain countries. A copy of MPack that we obtained was configured to trigger only on users from Russia, Ukraine and the United States. A user located in New Zealand that navigates to the malicious page would not be attacked. The PandaLabs Report shows an MPack installation that doesn't seem to be as selective triggering on users from different countries. Their statistic page shows successful attacks on users from Japan, Germany, Spain, United States, Romania, UK, Italy, France, China, Mexico and Canada.

The geolocation-dependent triggering could easily be extended into a more fine grained triggering mechanism to avoid specific networks. Web exploitation kits could create the illusion of a malicious server in a sheep skin for entities that find and assess malicious web servers (AV and securtity companies), but could continue to exhibit malicious behavior when accessed from outside these specific networks . For the attacker, it would lead to a greatly reduced risk of detection, while at the same time increasing the likelihood of continued operation of the malicious web server and therefore lead to continued financial gain for the attacker.

All web exploitation kits record information about the clients in a statistics database. An example of the effects statistics, shown in Figure 3, makes targeting easier. It provides attackers with information to determine whether their attacks are still successful. Do old attacks, such as the MDAC attack on Windows 2000, still lead to successful exploitations? If so, an attacker might refrain from upgrading to newer and more expensive exploits. If the success rate of an exploit suddenly drops, the attacker can react accordingly. Browser statistics and the subsequent trends allow the attacker to purchase exploits that provide the highest return on investment.

Geolocation information could potentially be used for follow-up attacks in the regions that were most successful. Successful rates in a specific region indicate that users are less security aware, do not use the latest patches, etc. and therefore will be prone to future attacks. These attacks are not limited to drive-by-downloads, but could be in the area of phishing, attacks on servers, SPAM, etc. These are the low hanging fruits attackers will go for first.

EXPLOIT SERVERS

In our previous KYE study, we observed that malicious web pages usually do not host the attack code directly. Rather, the attack code is being imported onto the "front-end" page (for example, via iframes), or the user is redirected to the attack code, as shown in Figure 6. IcePack and MPack confirm that this is common practice as it explicitly supports this structure in the statistics the tool collects. Every front-end page sends information about itself to the exploit server via the HTTP Referrer header, which is then recorded by the IcePack and MPack tool and visible on the statistics page as shown in Figure 3 and Figure 5. The attacker, as a result, can determine which front-end pages drive the most traffic to the exploit server and, equipped with this information, "marketing campaigns" or specific hacking attempts can be focused to effectively increase the attack success rate.

Usually the administrators of these front-end pages are not even aware that they are hosting code that includes or redirects to malicious content. These front-end servers might have fallen to victim to an attack themselves in which the attacker modifies the pages or, more covertly, performs man-in-the-middle attacks, such as ARP spoofing [11], to include the malicious content. In the case of ARP spoofing, the administrator of the web page cannot find any direct evidence of malicious pages on the web site, but users are attacked nevertheless. Tools that allow an attacker to perform automated ARP spoofing, such as HTool and zxarps [12], have been reported to be bundled with the MPack web exploitation kits, although we ourselves did not obtain a copy of such a tool.

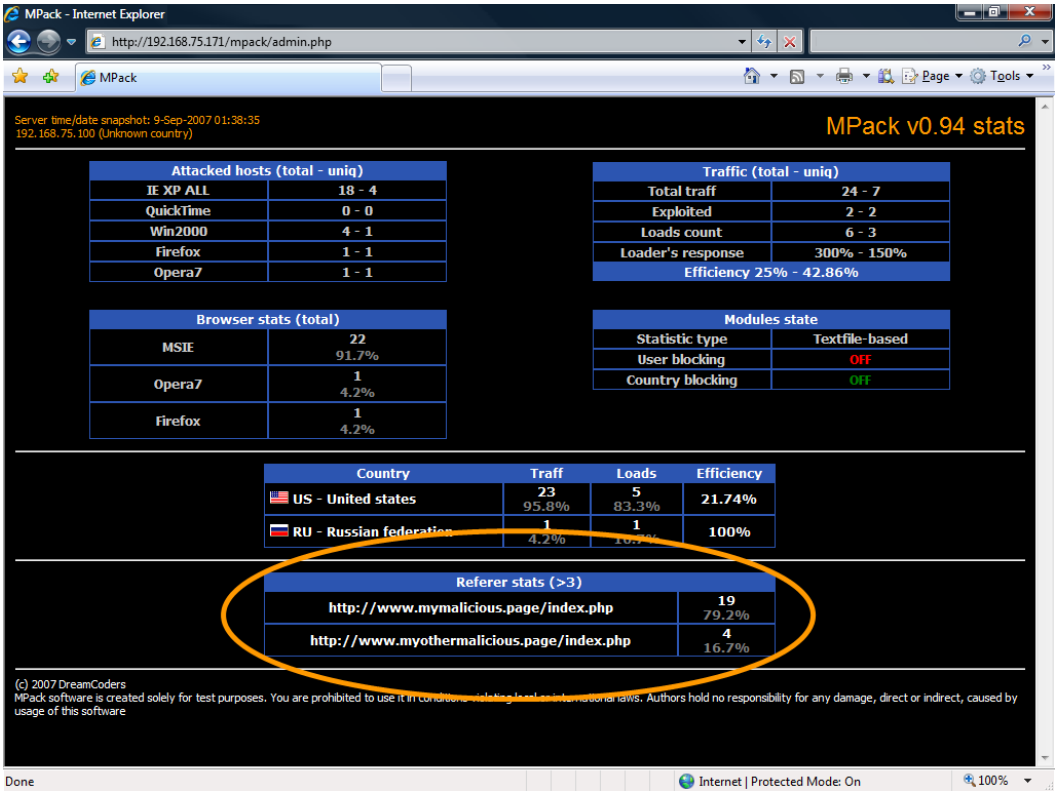

Figure 5 - Administrative Interface - Referer [sic] Stats

[11] http://www.avertlabs.com/research/blog/index.php/2007/10/04/arp-spoofing-is-your-web-hosting-service-protected/
[12] http://www.teamfurry.com/wordpress/2007/08/29/zxarps/#more-157

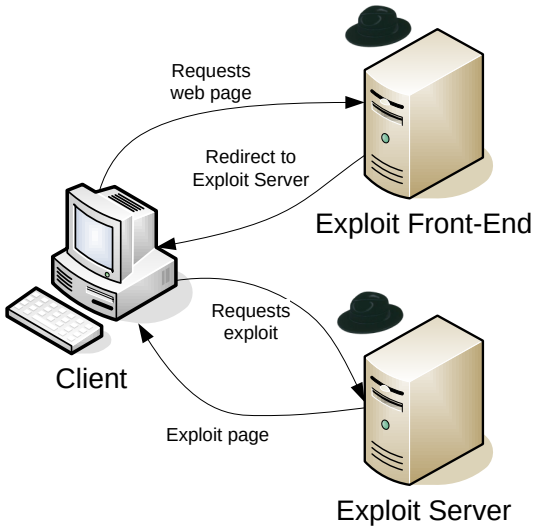

Figure 6 - Redirect to Exploit Server

Knowledge about these exploit servers is of high value. SANS Internet Storm Center identified a MPack attack in the middle of 2007 that showed 10,000 referral domains [13], [14]. This means that 10,000 web pages pointed to one exploit server. That is a high number. A live MPack server we found on the web (http://www.cordon.ru/mp/admin.php), "only" listed 504 referer URLs. A security administrator that is able to identify these servers will be able to greatly reduce the risk of infection by simply blacklisting these central exploit servers, effectively defusing risk of infection of the many front-end web pages that users potentially could come across.

Identification of these servers is simple. Since the exploitation kits consists of various PHP pages that host specific content, one can easily find exploit servers using a low interaction client honeypot, such as HoneyC. We performed such a search on several thousand known malicious URLs (sourced from stopbadware.org and mvps.org). For each URL, we checked for the existence of an admin.PHP page (the page to the administative console of MPack v0.94), as well as the specific string "All activity is being monitored", which can be found on these PHP pages. We were able to identify several MPack servers as shown in Table 2. Note that a search of this type was performed within a few hours on an average desktop PC using our low interaction client honeypot, HoneyC. The Snort signature used, which can also be used in the Snort Intrusion Detection System, is shown in Figure 7.

```
alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"Access to MPack v0.94 web exploitation kit
administrative console" flow:from_server,established; uricontent:"/admin.php"; content:"All activity is
being monitored"; reference:url,
http://blogs.pandasoftware.com/blogs/images/PandaLabs/2007/05/11/MPack.pdf; classtype:bad-unknown;
sid:TBD; rev:1;)
```

Figure 7 - Snort Signature MPack 0.94

[13] http://isc.sans.org/diary.html?storyid=3015
[14] http://ddanchev.blogspot.com/2007/06/massive-embedded-web-attack-in-italy.html

Search engine queries can also be used. Apparently a simple query that checks for `intitle: "Web Attacker Control"` used to result in several WebAttacker stats pages on Google [15]. In October 2007, however, Google doesn't return any results; and Live Search only returns one defunct instance of WebAttacker. Since search engines adhere to robots' directives and do not index standalone pages, the coverage is likely to be low and a low interaction client honeypot would be the preferred method of identifying a exploit server.

Hosts MPack (password protected stats page)
zz3b.info
xxxmovies.dip.jp
www.sys-browser.com
www.schirmsteher.de
www.mypornoxxx.com
www.gaba.bbchotnews.com
www.dom2.us
www.911traff.ws
www.911traff.info
www.555traff.ws
www.555traff.net
vorlagen-herunterladen.info
transfersarea.info
superengine.cn
stepbystepbg.org
soft.my-webs.org
sealhome.ru
redhotsocks.org
rallyesimages.ch
quote.a44.org.ua
qaq-tv.com
pop3mailers.info
pakk.jino-net.ru
oiuytr.jino-net.ru
nudeteens.in
nesmotri.com
mpack.redirfeed.com
mpack.phpnet.us
mazaing.com

Table 2 - Identified MPack Hosts

[15] http://ddanchev.blogspot.com/2007_02_01_archive.html

OBFUSCATION

Obfuscation is a mechanism to hide attacks from static detection tools, which use signatures to match against a known malicious string. Obfuscation causes the appearance of the malicious string to change therefore evading these detection tools. It is a technique we frequently encounter and something that is supported by the IcePack and MPack tool. With access to the MPack code, we are able to take a deeper look at how obfuscation is applied and whether there are weaknesses in the obfuscation routine.

Attack pages provided by MPack are obfuscated. The decryption routine executes three time before the attack page is in a state in which the browser can execute the contained attack code. A sample of the various obfuscation functions is shown in Figure 8, which leads to an obfuscated attack page as shown in Figure 9. It is obvious that a simple string matching algorithm would be unable to match on the attack code if this content changes upon every request.

```
00: function encrypt2($content)
01: {
02:   $table = "0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ_@";
03:   $xor = 165;
04:   $table = array_keys(count_chars($table, 1));
05:   $i_min = min($table);
06:   $i_max = max($table);
07:   for ($c = count($table); $c > 0; $r = mt_rand(0, $c--))
08:   array_splice($table, $r, $c - $r, array_reverse(array_slice($table, $r, $c - $r)));
09:   $len = strlen($content);
10:   $word = $shift = 0;
11:   for ($i = 0; $i < $len; $i++)
12:   {$ch = $xor ^ ord($content[$i]);
13:    $word |= ($ch << $shift);
14:    $shift = ($shift + 2) % 6;
15:    $enc .= chr($table[$word & 0x3F]);
16:    $word >>= 6;
17:    if (!$shift) { $enc .= chr($table[$word]); $word >>= 6; }}
18:    if ($shift)
19:    $enc .= chr($table[$word]);
20:   $tbl = array_fill($i_min, $i_max - $i_min + 1, 0);
21:   while (list($k,$v) = each($table))
22:   $tbl[$v] = $k;
23:   $tbl = implode(",", $tbl);
24:   $fi = ",p=0,s=0,w=0, t=Array({$tbl} )";
25:   $f  = "w|=(t[ x.charCodeAt(p++)-{$i_min}])<<s;";
26:   $f .= "if(s){r+=String.fromCharCode({$xor}^w&255);w>>=8;s-=2}else{s=6}";
27:   $r = "<script language=JavaScript>";
28:   $r.= "function dc(x){";
29:   $r.= "var l=x.length,i,j,r,b=(4096/4){$fi};";
30:   $r.= "for(j= Math.ceil(l/b);j>0;j--){r=''; for(i=Math.min(l,b);i>0;i--,l--
{{$f}}document.write(r)}";
31:   $r.= "}dc(\"{$enc}\")";
32:   $r.= "</script>";
33:   $r2 = "document.write(
unescape('%3C%73%63%72%69%70%74%3E%20%0D%0A%66%75%6E%63%74%69%6F%6E%20%7A%58%28%73%29%0D%0A%7B%20%76%61
%72%20%73%31%3D%20%75%6E%65%73%63%61%70%65%28%20%73%2E%73%75%62%73%74%72%28%30%2C%20%73%2E%6C%65%6E%67%
74%68%2D%31%29%29%3B%20%20%76%61%72%20%74%3D%27%27%3B%66%6F%72%28%69%3D%30%3B%69%3C%73%31%2E%6C%65%6E%6
7%74%68%3B%69%2B%2B%29%20%74%2B%3D%53%74%72%69%6E%67%2E%66%72%6F%6D%43%68%61%72%43%6F%64%65%28%20%73%31
%2E%63%68%61%72%43%6F%64%65%41%74%28%69%29%2D%20%73%2E%73%75%62%73%74%72%28%73%2E%6C%65%6E%67%74%68%2D%
31%2C%20%31%29%29%3B%20%0D%0A%64%6F%63%75%6D%65%6E%74%2E%77%72%69%74%65%28%75%6E%65%73%63%61%70%65%28%7
4%29%29%3B%20%7D%0D%0A%3C%2F%73%63%72%69%70%74%3E')); zX('".encodezTxt($content)."');";
34:   $r2 = "<Script Language='JavaScript'>".$r2."</Script>"; // [\\'\"]
35:   $r2 = "<script language=javascript>document.write(unescape(\"" .escape($r2). "\"))</script>";
36:   return $r2;
47: }
```

Figure 8 - MPack Obfuscation Routine

However, MPack v0.94 currently contains a major bug in its obfuscation routine: it produces unchanged obfuscated output on the same attack page (the bug is located on line 3 of Figure 8: instead of 165, it should be a random number). Even if the bug is fixed, there seems to be a finite number of obfuscated pages the current version of MPack could generate, and therefore static matching does fall into the realm of possibility on the attack page. Alternatively, several efforts attempt to decrypt obfuscated JavaScript and statically analyze the code in the clear [16], [17] or the code's behavior within a JavaScript engine [18].

```
<script
language=javascript>document.write(unescape("%3CScript%20Language%3D%27JavaScript%27%3Edocument.write%2
8%20unescape%28%27%253C%2573%2563%2572%2569%2570%2574%253E%2520%250D%250A%2566%2575%256E%2563%2574%2569
%256F%256E%2520%257A%2558%2528%2573%2529%250D%250A%257B%2520%2576%2561%2572%2520%2573%2531%253D%2520%25
75%256E%2565%2573%2563%2561%2570%2565%2528%2520%2573%252E%2573%2575%2562%2573%2574%2572%2528%2530%252C%
2520%2573%252E%256C%2565%256E%2567%2574%2568%252D%2531%2529%2529%253B%2520%2520%2576%2561%2572%2520%

...

%252A5I%252A5F%252A5I%252A5F%252A8H%252A7Kxhwnuy%252A8J%252A5I%252A5F%252A8H%252A7Kmjfi%252A8J%252A5I%2
52A5F%252A8Hgti%257E%252A75tsqtfi%252A8I%252A77xyfwy%252A7%253D%252A7%253E%252A77%252A8J%252A5I%252A5F%
252A8Hin%257B%252A75ni%252A8I%252A77r%257Ein%257B%252A77%252A8J%252A8H%252A7Kin%257B%252A8J%252A5I%252A
5F%252A8H%252A7Kgti%257E%252A8J%252A5I%252A5F%252A8H%252A7Kmyrq%252A8J%252A5I%252A5F%252A8Hnkwfrj%252A7
5%257Cniym%252A8I6%252A75mjnlmy%252A8I6%252A75gtwijw%252A8I5%252A75kwfrjgtwijw%252A8I5%252A75xwh%252A8I
%252A77myyu%252A8F%252A7K%252A7Kfqqmnlm2inxfgqji3twl%252A7Khtzsyjw%252A7Knsij%257D3umu%252A77%252A8J%25
2A8H%252A7Knkwfrj%252A8J5%27%29%3B%3C/Script%3E"))</script>
```

Figure 9 - Obfuscated Attack Page (Snippet)

On top of this obfuscation, however, additional obfuscation is likely to be applied to the front-end page that imports the page from the exploit server. These front-end pages do not merely place a static iframe on the page that imports the exploit. Rather, obfuscated JavaScript snippets append the iframe (via the document.write method) to the page once it is opened. Since this technique is independent of the web exploitation kit, it might even been applied with the earlier web exploitation kits that did not support obfuscation.

In addition, there are other means to determine whether a web exploitation kit is involved in an attack. A successful attack by a particular web exploitation kit seems to cause similar state changes on the client machine (see Appendix A for a complete list of state changes encountered on the URL http://www.keithjarrett.it, which was attacked by an MPack server). As such, simply reviewing the state changes that are caused on the client, one is able to infer what web exploitation kit was used in the attack. For example, reviewing the MPack attack page we observe that a file with the name "sys" plus four random characters and ".exe" is always pushed to C:\. Matching on this behavioral signature of this specific file write event among the state changes yields an identification of MPack attacks. From the 306 malicious URLs we identified in our KYE study, we were able to identify 13 URLs (or 4.24%) that utilized MPack using these behavioral signatures:

[16] http://www.ukhoneynet.org/2007/07/18/new-javascript-tool-released/
[17] http://cansecwest.com/slides07/csw07-nazario.pdf
[18] http://www.secureworks.com/research/tools/caffeinemonkey.html

URL
http://www.sexyclips.org/media.php?clip=one_night_in_paris_spoof.html
http://mashathumbs.com/
http://www.forumsplace.com/page_not_found.php
http://www.keithjarrett.it/
http://www.aerosmith.com/
http://www.versiontwo.org/old/archives/2003/05/jessica_lynch_r.html
http://www.spacefellowship.com/News/?p=1824
http://www.globalwarmingbar.com/
http://www.dalekeiger.com/?p=133
http://77kelvin.a55.nthosting.ru/
http://www.PoochTV.com
http://www.forumwz.org/archive/index.php/f-14.html
http://www.cracks.vg/cracks/The_Sims_2_Virtual_Disk_111912.html

Table 3 - MPack URLs

We also discovered a static iframe as part of the attack page in Mpack that attempted to import an additional page "/counter/index.php" from a the remote web server "allhigh.org". If this was indeed included by the creators of MPack, the purpose of it might be to count the number of times their framework is used or to potentially steal a customer for themselves. The path name indicates that it is used for counting, but because the URL is not live anymore we can not assure any explanation. Either way, if such static iframes are used, identification of a web exploitation kit would be straightforward: simply monitor DNS lookups to the specific host name. If a DNS lookup of allhigh.org is observed, that would be the indicator that a client attack by a web exploitation kit MPack was attempted.

CONCLUSIONS

In this paper, we have taken a behind-the-scenes look at malicious web servers that launch drive-by-downloads. We analyzed specific aspects of several web exploitation kits that allowed us to reexamine several questions that we were unable to answer with our previous study on malicious web servers. The web exploitation kits provided us with valuable insight on how malicious web servers operate. (For comprehensive analysis of MPack and WebAttacker we refer to existing reports [19],[20].) There are several conclusions we draw as a result of our analysis with respect to future studies and client honeypot technology:

[19] http://blogs.pandasoftware.com/blogs/images/PandaLabs/2007/05/11/MPack.pdf
[20] http://www.websense.com/securitylabs/blog/blog.php?BlogID=94

First, the appearance of web exploitation kits have a positive effect on identification of malicious web servers. Since they are tools that can be relatively easily obtained and deployed, it has a homogenizing effect on the malicious web server landscape (4.24% of the URLs encountered in our KYE study used MPack; WebAttacker apparently composed 32% of reported exploits in June 2006 [21]). Characteristics about the tool and the malicious content it serves can currently be identified and matched upon. We demonstrated that we can easily identify MPack servers using our low interaction client honeypot HoneyC as well as behavioral attack signatures. Whether attackers will invest energy and resources to "fix" these weaknesses remains to be seen.

Exploit servers are high value identification targets. Because they are used for numerous front-end pages, it is not likely that they will disappear. Rather, it is suspected that attackers will frequently update these exploit servers, which allows security researchers to observe the latest attacks. All web exploitation kits reviewed allow attackers to purchase upgrades, which will likely appear on the same machines.

On the other hand, the web exploitation kit MPack showed us that our identification of malicious web servers with high interaction client honeypot, such as Capture-HPC, has some limitations. First, the IP tracking functionality throws a wrench in the works of client honeypots and is likely to result in many false negatives if not addressed appropriately. A study into the magnitude of this technique and adjustments into the client honeypot approach are needed. It is likely that distributed client honeypots will become a necessity in the near future. Second, geolocation-dependent triggering is something we had not yet considered. Depending on what countries are prime targets, it might also result in a large number of false negatives. A comparitive study finding malicious servers using client honeypots in different physical locations will be necessary to assess the magnitude of the problem.

The web exploitation kits also showed us that other client applications/plug-ins are targeted. This is bad news for the end user whose many client vulnerabilities are now being actively attacked and comparing supported attacks of WebAttacker, MPack and IcePack seems to indicate a trend. As a result, client honeypots would require more complex configurations in search of malicious servers. In our previous study, we likely missed numerous attacks. With the insight provided by web exploitation kits, we are planning to address these gaps in future studies to come.

ACKNOWLEDGEMENTS

We would like to thank the following people:

- David Watson of the UK Honeynet Project (reviewer, provider of MPack v0.94)
- Raul Siles of the Spanish Honeynet Project (reviewer)
- Dave Dittrich of the Honeynet Project (reviewer)
- Thorsten Holz of the German Honeynet Project (reviewer)
- Mark Ryan del Moral Talabis from the Hawaii Honeynet Project (reviewer)

[21] http://explabs.com/about/mediaCenter/pr_071006_01.asp

FURTHER READING

For the interested reader, we provide references to additional papers/articles on MPack, malicious web servers and client honeypots:

- Seifert, C., Steenson, R., Holz, T., Bing, Y. and Davis, M.A. Know Your Enemy: Malicious Web Servers, The Honeynet Project, 2007, Available from http://www.honeynet.org/papers/mws/; accessed on 25 September 2007.
- The Honeynet Project. Know Your Enemy: Fast-Flux Service Networks, The Honeynet Project, 2007, Available from http://www.honeynet.org/papers/ff/; accessed on 25 September 2007.
- Seifert, C., Welch, I. and Komisarczuk, P., HoneyC - The Low-Interaction Client Honeypot. in NZCSRCS, (Hamilton, 2007), Available from http://www.mcs.vuw.ac.nz/~cseifert/blog/images/seifert-honeyc.pdf; accessed on 10 September 2006.
- Provos, N., McNamee, D., Mavrommatis, P., Wang, K. and Modadugu, N., The Ghost in the Browser: Analysis of Web-based Malware. in HotBots'07, (Cambridge, 2007), Usenix.
- Provos, N. and Holz, T. Client Honeypots. in Virtual Honeypots: From Botnet Tracking to Intrusion Detection, Addison Wesley Professional, Upper Saddle River, NJ, 2007, 231-272.

Appendix A

MPACK STATE CHANGES

Monitor	Action	Actor	Action parameter
file	Write	C:\Program Files\Internet Explorer\IEXPLORE.EXE	C:\syswcon.exe
process	Created	C:\Program Files\Internet Explorer\IEXPLORE.EXE	C:\syswcon.exe
file	Write	C:\syswcon.exe	C:\WINDOWS\system32\drivers\uzcx.exe
process	Created	C:\syswcon.exe	C:\WINDOWS\system32\drivers\uzcx.exe
process	Terminated	C:\Program Files\Internet Explorer\IEXPLORE.EXE	C:\syswcon.exe
registry	SetValueKey	C:\WINDOWS\system32\drivers\ uzcx.exe	HKCU\Software\ewrew\uzcx\main\cid
file	Write	C:\WINDOWS\system32\drivers\ uzcx.exe	C:\Documents and Settings\cseifert\ Local Settings\Temporary Internet Files\Content.IE5\OPUJWX63\ benupd32[1].exe
file	Write	C:\WINDOWS\system32\drivers\ uzcx.exe	C:\WINDOWS\benupd32.exe
process	Created	C:\WINDOWS\system32\drivers\ uzcx.exe	C:\WINDOWS\benupd32.exe
registry	SetValueKey	C:\WINDOWS\system32\drivers\ uzcx.exe	HKCU\Software\ewrew\uzcx\main\term
process	Created	C:\WINDOWS\benupd32.exe	C:\WINDOWS\benupd32.exe
file	Write		C:\Documents and Settings\cseifert\ Local Settings\Temp\clean_33d87.dll
process	Created	C:\WINDOWS\benupd32.exe"	C:\WINDOWS\system32\regsvr32.exe
registry	SetValueKey	C:\WINDOWS\explorer.exe	HKLM\SYSTEM\ControlSet001\Services\ ldrsvc\Parameters\ServiceDll

ABOUT THE AUTHOR

Christian Seifert is a PhD candidate at Victoria University of Wellington, New Zealand, and is currently on a visiting scholar appointment at the University of Washington in Seattle, WA. His research is targeted at improving the detection accuracy and speed of client honeypots. Christian has an MS in software engineering with a focus on computer security from Seattle University, WA. Christian has been a member of the New Zealand Honeynet Project since April 2006.