virus BULLETIN Covering the global threat landscape

LOKIBOT: DISSECTING THE C&C PANEL DEPLOYMENTS

Aditya K Sood

First advertised as an information stealer and keylogger when it appeared in underground forums in 2015, LokiBot has added various capabilities over the years and has affected many users worldwide. LokiBot is deployed as a botnet, where a number compromised systems installed with the malware connect with command-and-control (C&C) servers in order to send stolen data and receive commands from the botnet operator.

LokiBot has been distributed via phishing campaigns that include malicious attachments or embedded URLs [1]. More recently it has also been found to hide its source code in image files [2], using the technique known as steganography. LokiBot installs itself via a downloaded zipped file, which is deleted (in order to avoid detection) once the system has been infected. The malware steals credentials from the compromised system. The stolen data is compressed and exfiltrated via an HTTP channel to a C&C panel.

In this research, we conducted an analysis of the URL structure of the LokiBot C&C panels and how these have evolved over time, concentrating on the C&C panel entry points. In this paper the 'entry point' refers to the web access point used by the botnet operator to manage the botnet. This is basically a PHP web-based C&C panel component that gives the botnet operator administrator capabilities. We also highlight the gate component that is used as an entry point for the bots to communicate and transmit data. The gate can be considered one of the primary components of the C&C panel design because it provides gateway and filtering functionalities. In the majority of cases the gate component resides on the same server as the C&C panel, but it can be configured or changed accordingly.

The aim of this research is to build intelligence for detection and prevention solutions including security analytics.

LOKIBOT C&C PANEL: CHARACTERISTICS

In this section we look at the characteristics of the LokiBot C&C panel. A number of pointers are provided below:

- The LokiBot C&C panel is designed to use HTTP protocol as its communication mechanism.
- The C&C panel is entirely developed using PHP. The LokiBot C&C panel v3.0 base is built using PHP, which is used in conjunction with C++ and C# (the malware is written in these languages).
- The LokiBot C&C panel consists of two main components: the main administrative panel used by the botnet operator to administer the botnet, and the gate component that provides filtering capabilities so that data received from the compromised systems can be examined and bots can be verified. Other components are developed to ease the handling and management of stolen data from the compromised machines. (The C&C panel components are discussed in detail in the next section.)
- The data exfiltrated from the compromised endpoints is sent to the C&C panel in a compressed format over HTTP. The data is received by the gate component, which validates the authenticity of the data by checking the identity of the bot before the data is processed by the backend database and retrieved by the main C&C panel for the botnet operator to use it.
- LokiBot transmits data in zipped format and data log files are decrypted using a custom encryption and decryption algorithm that is used in conjunction with a Base-64 encoding/decoding mechanism.
- The LokiBot C&C panel can be deployed with anti-automation mechanisms to restrict account cracking attempts over HTTP. For that, a CAPTCHA is supported by the C&C panel. Figure 1 shows an example of a LokiBot C&C panel with CAPTCHA implementation; Figure 2 shows an example of a LokiBot C&C panel without CAPTCHA implementation.

LOKIBOT C&C PANEL: COMPONENTS

The basic structure of the LokiBot C&C panel with all the related components is outlined in Table 1.

The LokiBot C&C panel uses a gate component [3], which is written in PHP. Listing 1 shows how the LokiBot gate component extracts the source IP of the bot from which the connection is initiated. The extracted and analysed headers from the incoming HTTP traffic are presented below:

- X-Forwarded-For (or X-Forwarded-IP) shows that the source IP address is behind a proxy or a load balancer.
- HTTP_CF_CONNECTING_IP shows that the source IP address is behind the Cloudflare Content Delivery Network (CDN).
- X-ProxyUser-IP shows that the source IP address is behind Google Services.
- X-Real-IP shows that the source IP address is behind a load balancer.

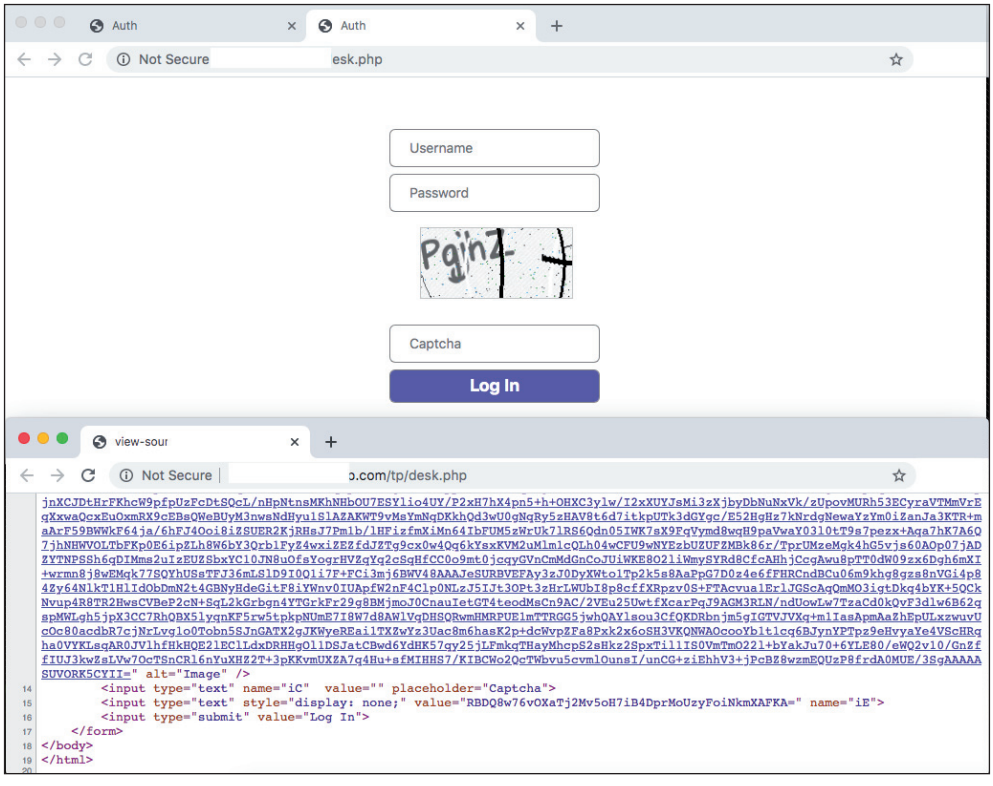

Figure 1: LokiBot C&C panel with CAPTCHA.

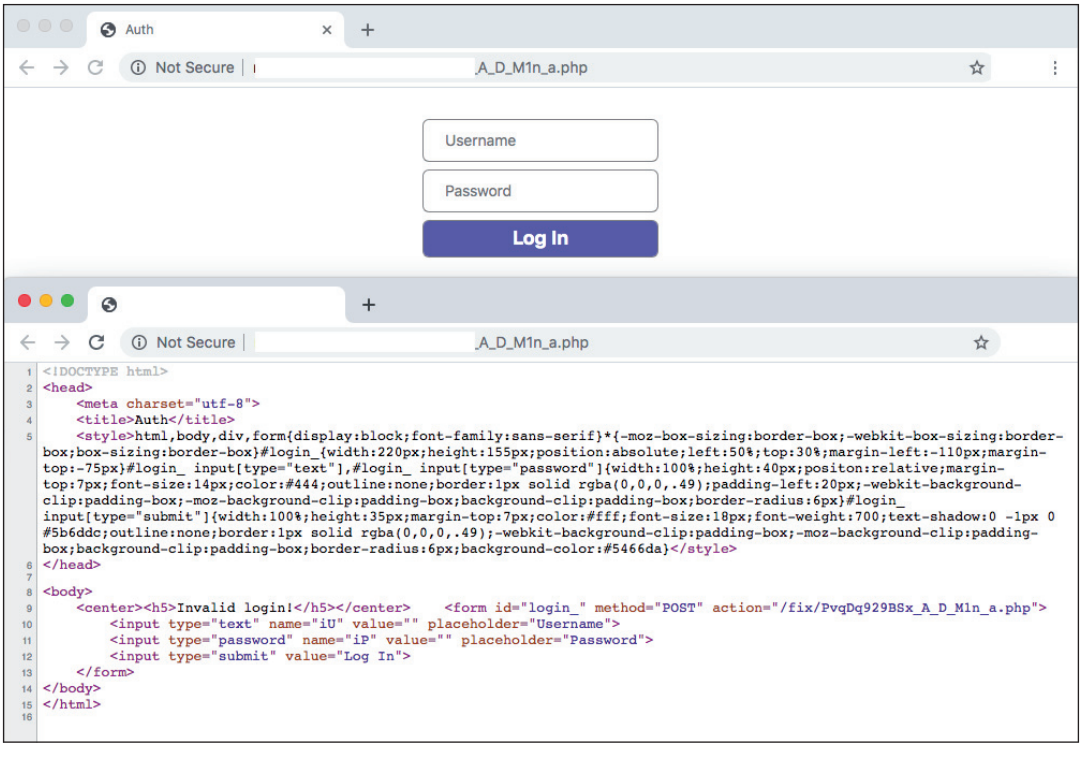

Figure 2: LokiBot C&C panel without CAPTCHA.

S. No	Component	Details
1	index.php	Main landing page of the C&C panel from where access is granted to the botnet operator.
2	gate.php	Intermediate proxy component that acts as an interface between the main C&C panel and the bots running on the compromised machines.
3	functions.php	Supporting functions such as error_reporting, base64Decrypt and traffic_decrypt are defined in this component.
4	install.php	Web component used to effectively deploy the C&C panel before spreading infections. The component installs the backend database, etc. to handle the stolen data, providing search capability, configuration tasks for the loader and others.
5	settings.php	This component configures the settings of the C&C panel including error handling, authentication, authorization, database configuration and others.
6	auth.php	This is the module deployed to configure the authentication for the C&C panel including how the gate authenticates itself to the C&C panel before storing stolen data in the database.
7	viewer.php	This component provides viewing capability to the botnet operator in the C&C panel so that data management is easy.
8	converter.php	This component provides converting capabilities to handle data in more efficient ways. For example, NetScapeToJson is used to convert cookies to JSON format.
9	search.php	This component provides a search capability to enable the botnet operator to search for and find specific data from the dump of stolen information stored in the backend database.
10	loader.php	This component is used to load the stolen data from the infected machines that is transferred by the gate component into the database and keep updating the records. This component also loads data from the database to the main C&C panel.
11	logs/	Folder used to store logs about stolen data and system-related errors.
12	tmp/	Temporary folder used to store the modules that are not required after installation of the C&C panel.
13	stealer/	Folder used to store a text file that defines the rules for the bot to steal data from specific URLs and domains. The file is passed to the bot running on the compromised system.
14	assets/	Folder used to store modules related to GeoIP, CSS for effective managing and laying out of data in the C&C panel.

Table 1: LokiBot C&C panel modules and components.

Listing 2 shows the basic authentication that can be configured to access the C&C panel. Form-based authentication is also supported.

Listing 3 shows how LokiBot decrypts the log files that are received from the compromised systems. The log file is decoded (or decrypted) using the 'base64Decrypt' function. The zipped file is extracted and passed to the 'TRAFFIC_DECRYPT' function, which decrypts the file to retrieve the stolen data. Once that operation is performed, a clean zip file containing the stolen data is created and then stored in the directory.

Listing 4 shows the support functions that are defined in the functions.php file. The 'base64Decrypt' and 'TRAFFIC_DECRYPT' functions highlight how the data decryption routines are handled in the C&C panel.

In the next section we discuss the result of the empirical analysis performed to analyse C&C panel URLs.

EMPIRICAL ANALYSIS: C&C DEPLOYMENTS

We looked into 1,960 different LokiBot C&C panel URLs deployed in real time. All the deployments of the C&C

```
<?php
ini_set('display_errors', 0);
ini_set('display_startup_errors', 0);
ini_set("allow_url_fopen", true);
ini_set("upload_max_filesize", "255M");
ini_set("post_max_size", "0");
ini_set("max_input_vars", "50000");
include 'database.php';
 require("functions.php");
$version = '3.0';
$xorKey="";
$outText='';
// Cloudflare support

if (isset($_SERVER["HTTP_CF_CONNECTING_IP"])) {
    $_SERVER['REMOTE_ADDR'] = $_SERVER["HTTP_CF_CONNECTING_IP"];
}
if (isset($_SERVER["X-Forwarded-For"])) {
    $_SERVER['REMOTE_ADDR'] = $_SERVER["X-Forwarded-For"];
}
if (isset($_SERVER["X-Forwarded-IP"])) {
    $_SERVER['REMOTE_ADDR'] = $_SERVER["X-Forwarded-IP"];
}
if (isset($_SERVER["X-ProxyUser-Ip"])) {
    $_SERVER['REMOTE_ADDR'] = $_SERVER["X-ProxyUser-Ip"];
}
if (isset($_SERVER["X-Real-IP"])) {
    $_SERVER['REMOTE_ADDR'] = $_SERVER["X-Real-IP"];
}
       $db = mysqli_connect(host,login,pass,base);
```

Listing 1: Module used by LokiBot for C&C authentication.

```
<?php
header('Cache-Control: no-cache, must-revalidate, max-age=0');
       $has_supplied_credentials = !(empty($_SERVER['PHP_AUTH_USER']) && empty($_
SERVER['PHP_AUTH_PW']));
       $is_not_authenticated = (
              !$has_supplied_credentials ||
              $_SERVER['PHP_AUTH_USER'] != $login ||
              md5($_SERVER['PHP_AUTH_PW'])   != $md5Password
       );
       if ($is_not_authenticated) {
              header('HTTP/1.1 401 Authorization Required');
              header('WWW-Authenticate: Basic realm="Access denied"');
              exit;
       }
?>
```

Listing 2: Module used by LokiBot for C&C authentication.

```
if(isset($_POST['logs'])){
        $logs = base64decrypt($_POST['logs']);
        $array = json_decode($logs, true);

        $key = base64decrypt($array['key']);
        $log = base64decrypt($array['log']);

        $zipa = TRAFFIC_DECRYPT($log, $key);
        $aaaaa = base64_encode($zipa);
                $zip = new ZipArchive();

                $zipname = 'logs.zip';
                $zip->open($zipname, ZipArchive::CREATE);
                        $zip->addFromString('logs.zip', base64_decode($aaaaa));


                $dira = 'logs.zip';
        $dir = $_SERVER['DOCUMENT_ROOT'].'/'.$dira;

                if(move_uploaded_file($tmp,$dir))

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

Listing 3: Module used by LokiBot for C&C authentication.

```
<?php
@error_reporting(0);
@set_time_limit(0);
@ini_set('max_execution_time', 0);
@ini_set('max_input_vars', 100000000);
@ini_set("memory_limit","500M");

function base64decrypt($str){
    $sub1 = str_replace("-", "+", $str);
    $sub2 = str_replace("_", "/", $sub1);
    $sub3 = str_replace(".", "=", $sub2);
    return base64_decode($sub3);
}

function TRAFFIC_DECRYPT($bytes, $key){
    $out = '';
    for($i = 0; $i < strlen($bytes); $i++){
        $out .= $bytes[$i] ^ $key[$i % 256];
    }

    return $out;
}
?>
```

Listing 4: Module used by LokiBot for C&C authentication.

panels were using PHP as the main component. The complete URLs comprised both domain names and IP addresses. Generally, IP addresses are used in C&C panels to avoid DNS queries so that DNS traffic can be avoided from the compromised endpoint. This way, the endpoints can connect directly with the C&C panel by initiating the connection to IP address. The data analysis was performed on the primary C&C panel component, i.e. the main entry PHP web page that is used by the botnet operator to administer the botnet.

Table 2 highlights the C&C components utilizing the PHP page as the entry point for the botnet operators to manage the LokiBot instances in the real world. Table 3 highlights the percentage layout of the LokiBot C&C entry points deployed in real time.

The details presented here highlight the different entry points that are configured for LokiBot C&C panel communication.

INFERENCES

1. Approximately 95% of LokiBot deployments in real time use 'PvqDq929BSx_A_D_M1n_a.php' as the main entry point.
2. The 'admin' in the string 'PvqDq929BSx_A_D_M1n_a.php' is represented as '_A_D_M1n_a.php' to avoid standard-level detections that analyse basic URL structure.
3. The other C&C entry points – 'desk.php', 'sand.php', 'omc.php', 'uMc.php', etc. – represent just 5% of the dataset chosen for analysis, which shows that an obfuscated string is preferred in the resource naming for the C&C entry point.
4. The majority of the LokiBot C&C deployments are configured over HTTP without TLS, i.e. a non-HTTPS channel is used for communication. As a result, all the communication can be seen over an

LokiBot C&C entry point	Server-side language: C&C panel	Usage
'PvqDq929BSx_A_D_M1n_a.php'	PHP	1,861
'pen.php'	PHP	31
'desk.php'	PHP	18
'omc.php'	PHP	12
'uMc.php'	PHP	17
'sand.php'	PHP	03
'Pvq.php'	PHP	12
'cs.php'	PHP	04
'loki.php'	PHP	02

Table 2: Deployed LokiBot C&C instances.

Lokibot C&C entry point	Percentage
'PvqDq929BSx_A_D_M1n_a.php'	94.95%
'pen.php'	1.58%
'desk.php'	0.92%
'omc.php'	0.61%
'uMc.php'	0.87%
'sand.php'	0.15%
'Pvq.php'	0.61%
'cs.php'	0.20%
'loki.php'	0.10%

Table 3: Percentage analysis of total instances of LokiBot C&C panels.

unencrypted channel. LokiBot does provide HTTPS support but it has to be configured explicitly.

5. From the compromised machines, the stolen data transmitted by the bot is received by the gate component first, which analyses the data to verify the authenticity of the bot. Once the bot identity is established, the stolen data is transmitted to the backend storage so that it can be analysed and accessed in the C&C panel.

CONCLUSION

Conducting an empirical analysis of LokiBot's C&C structure helps to build intelligence that can be used to enhance the detection and prevention efficacy of security solutions. It also helps to unearth the advancements in techniques used by the attackers to trigger infections and steal data.

REFERENCES

[1] Newly Discovered Infostealer Attack Uses LokiBot. Fortinet. https://www.fortinet.com/blog/threat-research/new-infostealer-attack-uses-lokibot.html.

[2] LokiBot malware now hides its source code in image files. ZDNet. https://www.zdnet.com/article/lokibot-information-stealer-now-hides-malware-in-image-files/.

[3] Sood, A.K.; Bansal, R. Prosecting the Citadel botnet – revealing the dominance of the Zeus descendent: part one. https://www.virusbulletin.com/virusbulletin/2014/09/prosecting-citadel-botnet-revealing-dominance-zeus-descendent-part-one.

Head of Testing: Peter Karsai

Security Test Engineers: Gyula Hachbold, Adrian Luca, Csaba Mészáros, Tony Oliveira, Ionuț Răileanu

Sales Executive: Allison Sketchley

Editorial Assistant: Helen Martin

Tel: +44 20 3920 6348 Email: editorial@virusbulletin.com

Web: https://www.virusbulletin.com/