Mozilla位于AWS上的0day RCE

Hulk / 2019-03-25 09:10:00 / 浏览数 2775 渗透测试 渗透测试 顶(0) 踩(0)

文章来源：https://blog.assetnote.io/bug-bounty/2019/03/19/rce-on-mozilla-zero-day-webpagetest/

前言

当你在渗透测试过程中使用Assetnote Continuous Security（CS，安全服务产品）时，该服务会识别部署在实例（比如AWS, GoogleCloud等）上的WebPageTest。WebPageTest是一款网站（前端）性能分析工具，使用该工具你可以测试给定URL或主机的性能状况。

用户可以通过修改WebPageTest的setting.ini文件来实现基本的安全认证，同时这可以杜绝匿名访问。但是Assetnote CS发现大部分的WebPageTest服务都没有开启认证机制。WebPageTest提供了一些网站测试的工具，利用这些工具攻击者可以发起服务端请求伪造攻击（简称为SSRF，W

在2017年11月，Assetnote CS发现了Mozilla位于AWS环境下的两个子域名：

- wpt-vpn.stage.mozaws.net
- wpt1.dev.mozaws.net

这两个都是处于实例中WebPageTest服务，并且没有认证机制。说实话，这是Assetnote CS第一次发现该服务还获得了赏金的。随后我与Mathias进行合作审计源代码，几个小时后我们就创建一个可以导致远程代码执行的攻击链条。

本次发现的漏洞是一个0day，那时我们与Mozilla和WebPageTest团队通力合作，成功修复该漏洞。提交该漏洞后，WebPageTest团队在2018年1月17日发布更新程序。

漏洞挖掘

我们发现的第一个有趣的事是可以上传和解压任意Zip文件，漏洞通过`/www/work/workdone.php`得已实现。这个脚本通过某些逻辑限制用户只能通过127.0.0.1来实施操作

```
...
!strcmp($_SERVER['REMOTE_ADDR'], "127.0.0.1")
...
```

我们稍后再讨论此处的问题。

还是在这个文件，我们发现了另一个潜在的风险向量。在上传任意的Zip文件后，文件将被解压到一个确定的位置：

/www/work/workdone.php 中133 - 136行：

```
if (isset($_FILES['file']['tmp_name'])) {
 ExtractZipFile($_FILES['file']['tmp_name'], $testPath);
 CompressTextFiles($testPath);
}
```

如果我们把IP伪造成127.0.0.1，那么我们利用此风险点似乎就可以获取代码执行权限。

然而事情并非是我们想象地那么简单，由于/www/work/workdone.php中321行处代码：

```
SecureDir($testPath);
```

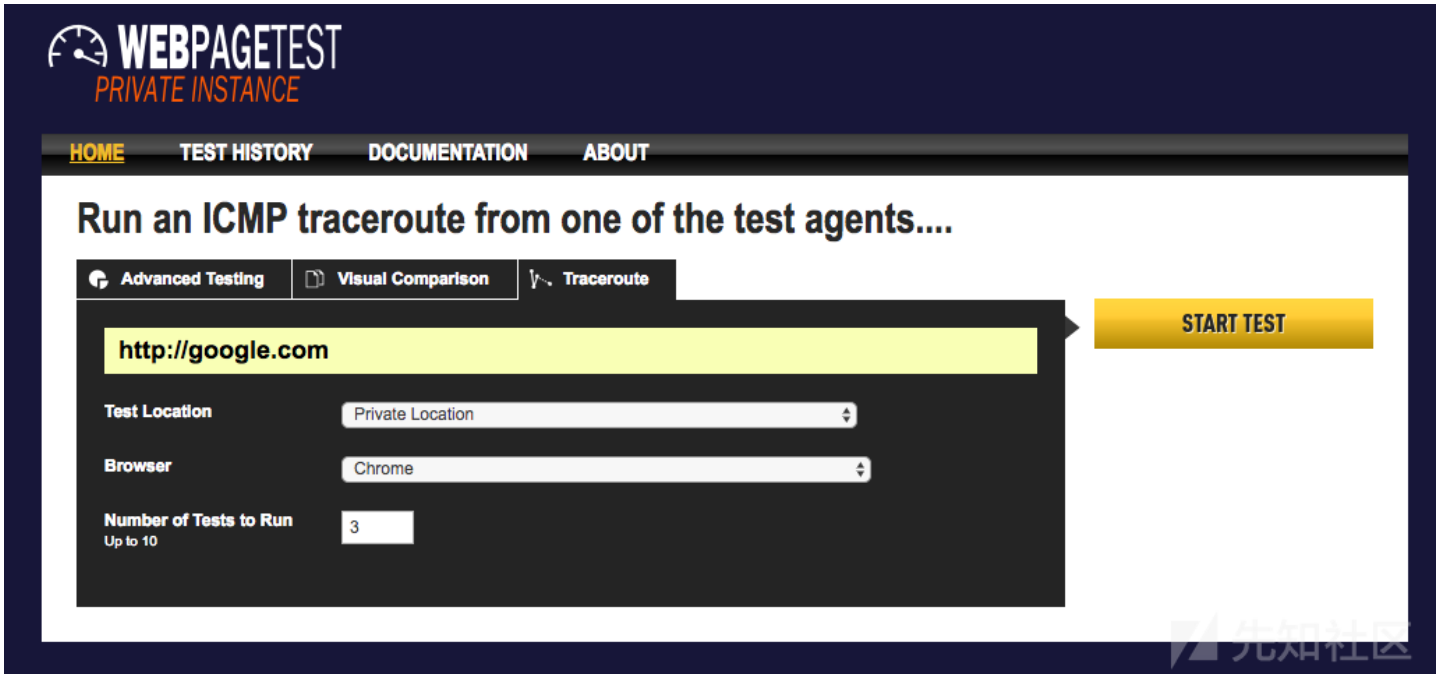
跟进`SecureDir`函数到/www/common_lib.inc中的2322 - 2347行处：

```
/**
* Make sure there are no risky files in the given directory and make everything no-execute
*
* @param mixed $path
*/
function SecureDir($path) {
   $files = scandir($path);
   foreach ($files as $file) {
       $filepath = "$path/$file";
       if (is_file($filepath)) {
           $parts = pathinfo($file);
           $ext = strtolower($parts['extension']);
           if (strpos($ext, 'php') === false &&
               strpos($ext, 'pl') === false &&
               strpos($ext, 'py') === false &&
               strpos($ext, 'cgi') === false &&
               strpos($ext, 'asp') === false &&
```

```
                strpos($ext, 'js') === false &&
                strpos($ext, 'rb') === false &&
                strpos($ext, 'htaccess') === false &&
                strpos($ext, 'jar') === false) {
                @chmod($filepath, 0666);
            } else {
                @chmod($filepath, 0666);    // just in case the unlink fails for some reason
                unlink($filepath);
            }
        } elseif ($file != '.' && $file != '..' && is_dir($filepath)) {
            SecureDir($filepath);
        }
    }
}
```

因为在代码运行流程中，SecureDir函数表现出一些延迟。这有可能导致一个可利用的竞争条件（漏洞），因为当PHP文件被解压到web服务器时不会立马被删除，所以用

攻击链的第一个预设条件非常简单，访问wpt-vpn.stage.mozaws.net的WebPageTest页面，对https://google.com进行Traceroute（跟踪路由）操作，此时我们获



在运行完traceroute后，WebPageTest 将我们重定向到另一个页面，该页面的URL包含测试ID：

- http://wpt-vpn.stage.mozaws.net/result/171124_GW_9/

但现在我们仍然需要某些方法来伪装成127.0.0.1，从而利用该脚本的漏洞函数。

利用以下的代码逻辑，我们可以实现伪装，/www/common.inc的70行处：

```
if (isset($_SERVER["HTTP_FASTLY_CLIENT_IP"]))
 $_SERVER["REMOTE_ADDR"] = $_SERVER["HTTP_FASTLY_CLIENT_IP"];
```

通过发送FASTLY-CLIENT-IP为127.0.0.1的请求头，我们可以远程设置$_SERVER["REMOTE_ADDR"]为任意内容。

组合攻击

综合上述的几个漏洞元素，我们能够通过两个Burp Intruder（入侵者）攻击，最终获取远程代码执行。

对于第一个Burp入侵者攻击，它用来上传含有恶意内容的Zip文件，另一个则是尝试与解压到系统中的PHP文件链接。我们的思路是将Burp Intruder（发包）线程提升到200以上，从而利用竞争条件漏洞。

其实我们可以使用请求效率更高的一些工具，比如说Turbo Intruder，这可能更容易实现攻击。

最终我们能够获取Mozilla的远程代码执行权限，见下图：

Results | Target | Positions | Payloads | Options

Filter: Showing all items

Request	Payload	Status	Error	Timeout	Length	Comment
1170	1170	200	☐	☐	93947	
938	938	200	☐	☐	93939	
0		404	☐	☐	469	
1	1	404	☐	☐	469	
2	2	404	☐	☐	469	
3	3	404	☐	☐	469	

Request | Response

Raw | Headers | Hex | HTML | Render

```
HTTP/1.1 200 OK
Date: Fri, 24 Nov 2017 19:19:05 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 93752
Connection: close
X-Powered-By: PHP/5.6.31
Vary: Accept-Encoding
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"><head>
<style type="text/css">
body {background-color: #fff; color: #222; font-family: sans-serif;}
pre {margin: 0; font-family: monospace;}
a:link {color: #009; text-decoration: none; background-color: #fff;}
a:hover {text-decoration: underline;}
table {border-collapse: collapse; border: 0; width: 934px; box-shadow: 1px 2px 3px #ccc;}
.center {text-align: center;}
.center table {margin: 1em auto; text-align: left;}
.center th {text-align: center !important;}
td, th {border: 1px solid #666; font-size: 75%; vertical-align: baseline; padding: 4px 5px;}
h1 {font-size: 150%;}
h2 {font-size: 125%;}
.p {text-align: left;}
.e {background-color: #ccf; width: 300px; font-weight: bold;}
.h {background-color: #99c; font-weight: bold;}
.v {background-color: #ddd; max-width: 300px; overflow-x: auto;}
.v i {color: #999;}
img {float: right; border: 0;}
hr {width: 934px; background-color: #ccc; border: 0; height: 1px;}
</style>
<title>phpinfo()</title><meta name="ROBOTS" content="NOINDEX,NOFOLLOW,NOARCHIVE" /></head>
<body><div class="center">
<table>
<tr class="h"><td>
<a href="http://www.php.net/"><img border="0"
src="
```

在我们报告此漏洞后，Bugzilla官方也发布了相关漏洞的公告。该公告包含具体的复现步骤，这对想要重现该漏洞的笔试者应该有所帮助。

该漏洞属于Mozilla漏洞悬赏项目，我们被奖励了500美金。

上一篇：多个网站的账户接管事例 下一篇：.NET高级代码审计（第三课）Fa...

1. 0 条回复

- 动动手指，沙发就是你的了！

目录