0ctf Wallbreaker Easy 详细 Writeup

rmb122 / 2019-03-28 08:49:00 / 浏览数 3071 安全技术 CTF 顶(0) 踩(0)

周末打了两天, 自闭 web 狗就做出来这一题, 题目质量真的是非常高, 学到了很多.

描述

```
Imagick is a awesome library for hackers to break `disable_functions`.
So I installed php-imagick in the server, opened a `backdoor` for you.
Let's try to execute `/readflag` to get the flag.
Open basedir: /var/www/html:/tmp/949c1400c8390865cb5939a106fec0b6
Hint: eval($_POST["backdoor"]);
```

提示

```
WALLBREAKER EASY
Ubuntu 18.04 / apt install php php-fpm php-imagick
```

第一眼看上去, 直接给了 webshell, 要求是执行根目录下的 `/readflag`,
先执行一波 `phpinfo()` 看看信息.

$_SERVER['SERVER_SOFTWARE']	nginx/1.14.0

System	Linux 789ff177e58f 4.15.0-46-generic #49-Ubuntu SMP Wed Feb 6 09:33:07 UTC 2019 x86_64	
Build Date	Feb 8 2019 14:54:22	
disable_functions	pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wifcontinued,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,pcntl_signal_get_handler,pcntl_signal_dispatch,pcntl_get_last_error,pcntl_strerror,pcntl_sigprocmask,pcntl_sigwaitinfo,pcntl_sigtimedwait,pcntl_exec,pcntl_getpriority,pcntl_setpriority,pcntl_async_signals,system,exec,shell_exec,popen,proc_open,passthru,symlink,link,syslog,imap_open,ld,mail	pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wifcontinued,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,pcntl_signal_get_handler,pcntl_signal_dispatch,pcntl_get_last_error,pcntl_strerror,pcntl_sigprocmask,pcntl_sigwaitinfo,pcntl_sigtimedwait,pcntl_exec,pcntl_getpriority,pcntl_setpriority,pcntl_async_signals,system,exec,shell_exec,popen,proc_open,passthru,symlink,link,syslog,imap_open,ld,mail

可以看到是 lnp 64 位环境, disable_function 如下

```
pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wifcontinued,pcntl_we
```

Imagick 信息如下

imagick module	enabled
imagick module version	3.4.3RC2
imagick classes	Imagick, ImagickDraw, ImagickPixel, ImagickPixelIterator, ImagickKernel
Imagick compiled with ImageMagick version	ImageMagick 6.9.7-4 Q16 x86_64 20170114 http://www.imagemagick.org
Imagick using ImageMagick library version	ImageMagick 6.9.7-4 Q16 x86_64 20170114 http://www.imagemagick.org
ImageMagick copyright	© 1999-2017 ImageMagick Studio LLC
ImageMagick release date	20170114
ImageMagick number of supported formats:	220
ImageMagick supported formats	3FR, AAI, AI, ART, ARW, AVI, AVS, BGR, BGRA, BGRO, BIE, BMP, BMP2, BMP3, BRF, CAL, CALS, CANVAS, CAPTION, CIN, CIP, CLIP, CMYK, CMYKA, CR2, CRW, CUR, CUT, DATA, DCM, DCR, DCX, DDS, DFONT, DNG, DPX, DXT1, DXT5, EPDF, EPI, EPS, EPS2, EPS3, EPSF, EPSI, EPT, EPT2, EPT3, ERF, FAX, FILE, FITS, FRACTAL, FTP, FTS, G3, G4, GIF, GIF87, GRADIENT, GRAY, GROUP4, H, HALD, HDR, HISTOGRAM, HRZ, HTM, HTML, HTTP, HTTPS, ICB, ICO, ICON, IIQ, INFO, INLINE, IPL, ISOBRL, ISOBRL6, JBG, JBIG, JNG, JNX, JPE, JPEG, JPG, JPS, JSON, K25, KDC, LABEL, M2V, M4V, MAC, MAGICK, MAP, MASK, MAT, MATTE, MEF, MIFF, MKV, MNG, MONO, MOV, MP4, MPC, MPEG, MPG, MRW, MSL, MTV, MVG, NEF, NRW, NULL, ORF, OTB, OTF, PAL, PALM, PAM, PATTERN, PBM, PCD, PCDS, PCL, PCT, PCX, PDB, PDF, PDFA, PEF, PES, PFA, PFB, PFM, PGM, PICON, PICT, PIX, PJPEG, PLASMA, PNG, PNG00, PNG24, PNG32, PNG48, PNG64, PNG8, PNM, PPM, PREVIEW, PS, PS2, PS3, PSB, PSD, PTIF, PWP, RADIAL-GRADIENT, RAF, RAS, RAW, RGB, RGBA, RGBO, RGF, RLA, RLE, RMF, RW2, SCR, SCT, SFW, SGI, SHTML, SIX, SIXEL, SPARSE-COLOR, SR2, SRF, STEGANO, SUN, TEXT, TGA, THUMBNAIL, TIFF, TIFF64, TILE, TIM, TTC, TTF, TXT, UBRL, UBRL6, UIL, UYVY, VDA, VICAR, VID, VIFF, VIPS, VST, WBMP, WMV, WPG, X, X3F, XBM, XC, XCF, XPM, XPS, XV, XWD, YCbCr, YCbCrA, YUV

题解

可以看到禁用了全部能直接执行程序的函数, 顺便还禁用了 mail, 不然的话可以直接用 LD_PRELOAD 来执行系统命令. 具体可以看这个项目.

接下来第一反应是通过 ghostscript 的 0day 打一波试试, 但尝试了几次全部都没有反应... 好吧没有报错太蛋疼了, 我们还是照着提示搭一波环境吧.

然后发现这里其实就真的跟提示一样, 全是最新的环境, 所以肯定是不存在已知的严重 0day 的...
同时, 可以看到在默认的配置文件中, 已经禁用了 ghostscript 的使用, 不能通过 gs 来命令执行.

```
$ cat /etc/ImageMagick-6/policy.xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE policymap [
<!ELEMENT policymap (policy)+>
<!ELEMENT policy (#PCDATA)>
<!ATTLIST policy domain (delegate|coder|filter|path|resource) #IMPLIED>
<!ATTLIST policy name CDATA #IMPLIED>
<!ATTLIST policy rights CDATA #IMPLIED>
<!ATTLIST policy pattern CDATA #IMPLIED>
<!ATTLIST policy value CDATA #IMPLIED>
]>
<policymap>
 <!-- <policy domain="resource" name="temporary-path" value="/tmp"/> -->
 <policy domain="resource" name="memory" value="256MiB"/>
 <policy domain="resource" name="map" value="512MiB"/>
 <policy domain="resource" name="width" value="16KP"/>
 <policy domain="resource" name="height" value="16KP"/>
 <policy domain="resource" name="area" value="128MB"/>
 <policy domain="resource" name="disk" value="1GiB"/>
 <policy domain="delegate" rights="none" pattern="URL" />
 <policy domain="delegate" rights="none" pattern="HTTPS" />
 <policy domain="delegate" rights="none" pattern="HTTP" />
 <!-- in order to avoid to get image with password text -->
 <policy domain="path" rights="none" pattern="@*"/>
 <policy domain="cache" name="shared-secret" value="passphrase" stealth="true"/>
 <!-- disable ghostscript format types -->
 <policy domain="coder" rights="none" pattern="PS" />
 <policy domain="coder" rights="none" pattern="EPI" />
 <policy domain="coder" rights="none" pattern="PDF" />
 <policy domain="coder" rights="none" pattern="XPS" />
</policymap>
```

看起来好像一切的都没了希望, 但这时突然想起, 有没有可能通过 Imagick 来达到 mail 函数类似的效果呢? 上面的 LD_PRELOAD
其实是劫持了启动进程这一行为, 也就是说, 如果我们能让 Imagick 调用外部进程, 我们完全可以不通过 mail 来执行系统命令.

Imagick 的底层是 ImageMagick, 这里就要说 ImageMagick 的 delegate 问题, ImageMagick 其实并没有实现所有文件格式的转换,
而是启动外部程序来进行转换, 就像上面的 ghostscript, 如果要将图片转换为 pdf, 就需要调用 ghostscript 来转换, 而这就会启动新的进程, 触发 LD_PRELAOD.

```
$ cat /etc/ImageMagick-6/delegates.xml
<!DOCTYPE delegatemap [
<!ELEMENT delegatemap (delegate)+>
<!ELEMENT delegate (#PCDATA)>
<!ATTLIST delegate decode CDATA #IMPLIED>
<!ATTLIST delegate encode CDATA #IMPLIED>
<!ATTLIST delegate mode CDATA #IMPLIED>
<!ATTLIST delegate spawn CDATA #IMPLIED>
<!ATTLIST delegate stealth CDATA #IMPLIED>
<!ATTLIST delegate thread-support CDATA #IMPLIED>
<!ATTLIST delegate command CDATA #REQUIRED>
]>
<delegatemap>
<!-- ■■■■■ -->
 <delegate decode="bmp" encode="wdp" command="/bin/mv &quot;%i&quot; &quot;%i.bmp&quot;; &quot;JxrEncApp&quot; -i &quot;%i.bmp
 <delegate decode="ppt" command="&quot;soffice&quot; --convert-to pdf -outdir `dirname &quot;%i&quot;` &quot;%i&quot; 2&gt; &q
 <delegate decode="pptx" command="&quot;soffice&quot; --convert-to pdf -outdir `dirname &quot;%i&quot;` &quot;%i&quot; 2&gt; &
 <delegate decode="ps:alpha" stealth="True" command="&quot;gs&quot; -sstdout=%%stderr -dQUIET -dSAFER -dBATCH -dNOPAUSE -dNOPR
 <delegate decode="ps:cmyk" stealth="True" command="&quot;gs&quot; -sstdout=%%stderr -dQUIET -dSAFER -dBATCH -dNOPAUSE -dNOPRO
<!-- ■■■■■ -->
</delegatemap>
```

可以看到有很多的 delegate, 在进行这些格式的编码以及解码时, ImageMagick 会调用这些外部程序. 但是这些外部程序大部分其实都不自带, 得自己安装,
不会起新的进程, 但是可以注意到这个特殊的格式,

<delegate decode="bmp" encode="wdp" command="/bin/mv "%i" "%i.bmp"; "JxrEncApp" -i "%i.bmp&

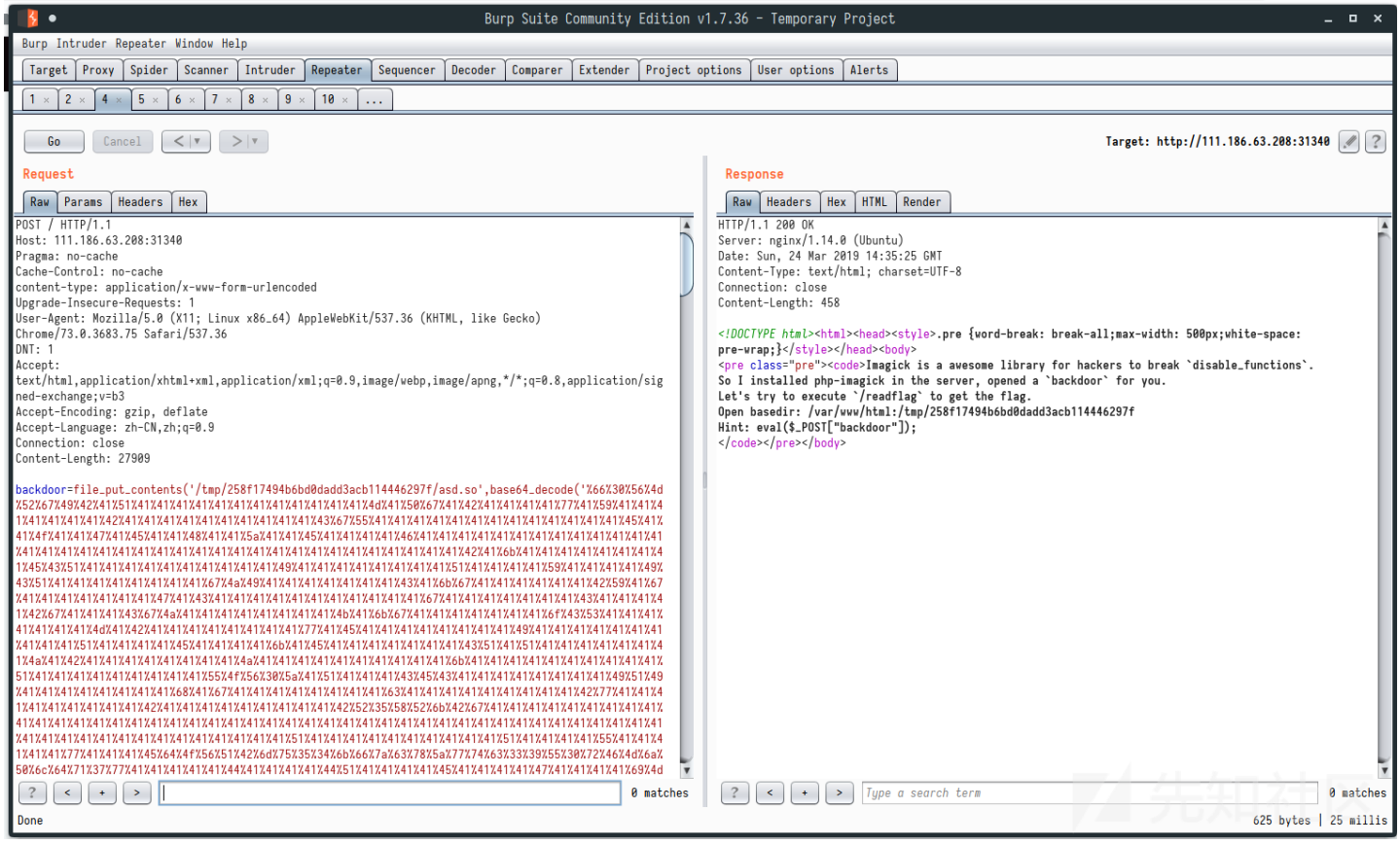
其中有系统自带的 mv, 这肯定是能执行成功的, 我们只要通过转换格式就能调用这个 delegate.
通过 strace 也可以发现确实调用了 mv, 即使因为 JxrEncApp 不存在导致图片转换失败, 但只要因为 mv 调起了新进程, 我们就能执行任意命令.

```
$ strace php -a 2&>1
Interactive shell

php > $a = new Imagick();
php > $a->readImage('123.png');
php > $a->writeImage('sad.wdp');

Warning: Uncaught ImagickException: delegate failed `/usr/bin/mv '%i' '%i.bmp'; 'JxrEncApp' -i '%i.bmp' -o '%o.jxr'; /usr/bin/mv '%i.bmp' '%i'; /usr/bin/mv '
%o.jxr' '%o'' @ error/delegate.c/InvokeDelegate/1867 in php shell code:1
Stack trace:
#0 php shell code(1): Imagick->writeimage('sad.wdp')
#1 {main}
  thrown in php shell code on line 1

php >
```

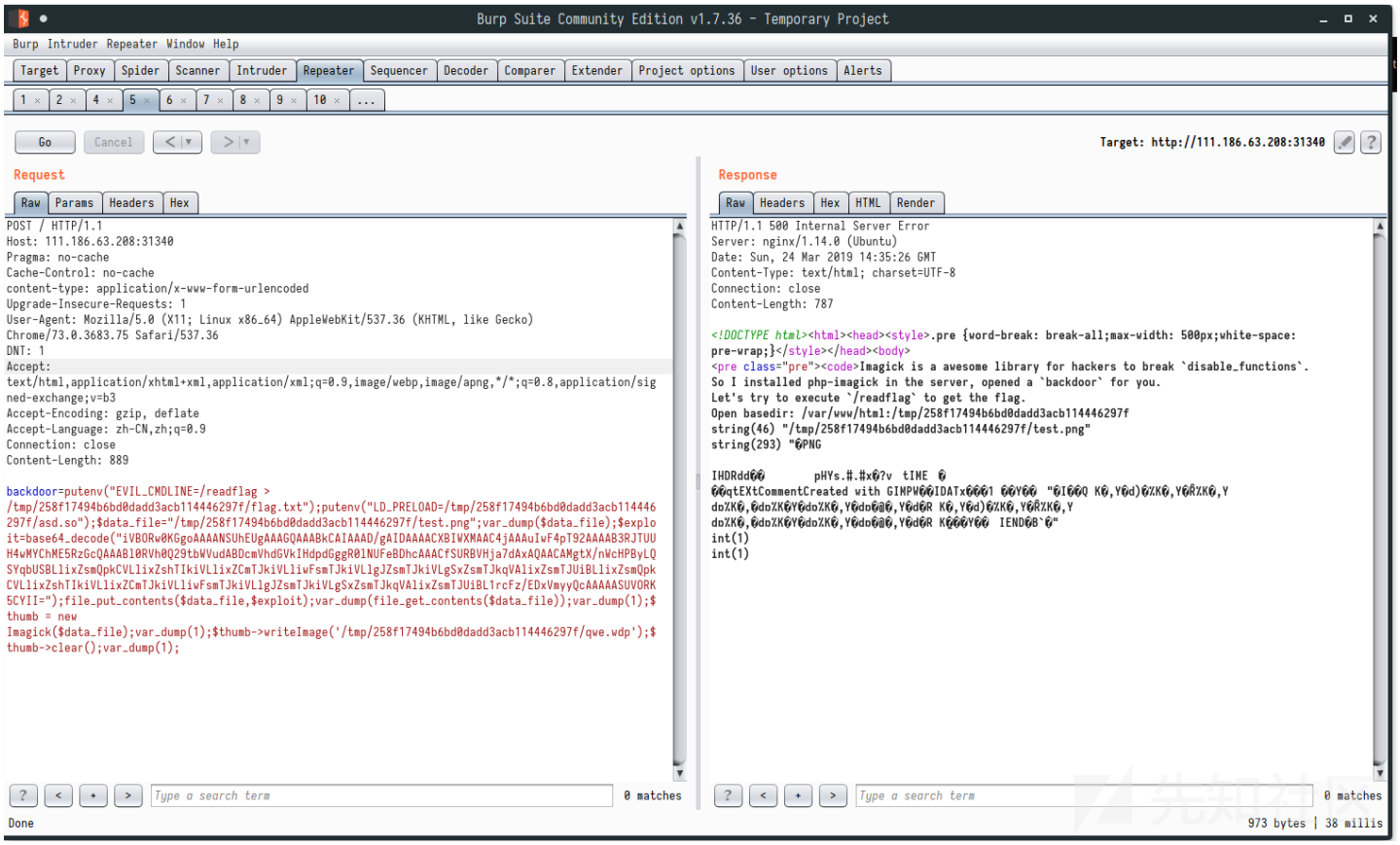
```
$ cat 1 | grep mv
/usr/bin/mv: 无法获取'/tmp/magick-28780fr3eBPd11YRD.jxr' 的文件状态(stat)/usr/bin/mv: 无法获取'/tmp/magick-28780fr3eBPd11YRD.jxr' 的文件状态(stat): 没有那个
文件或目录
write(2, "PHP Warning:  Uncaught ImagickEx"..., 348write(2, "PHP Warning:  Uncaught ImagickEx"..., 348PHP Warning:  Uncaught ImagickException: delegate faile
d `/usr/bin/mv '%i' '%i.bmp'; 'JxrEncApp' -i '%i.bmp' -o '%o.jxr'; /usr/bin/mv '%i.bmp' '%i'; /usr/bin/mv '%o.jxr' '%o'' @ error/delegate.c/InvokeDelegate/18
67 in php shell code:1
PHP Warning:  Uncaught ImagickException: delegate failed `/usr/bin/mv '%i' '%i.bmp'; 'JxrEncApp' -i '%i.bmp' -o '%o.jxr'; /usr/bin/mv '%i.bmp' '%i'; /usr/bin
/mv '%o.jxr' '%o'' @ error/delegate.c/InvokeDelegate/1867 in php shell code:1
```

这样, 就可以通过 `ImageMagick` 来触发新进程的产生, 并通过修改 `LD_PRELOAD` 的方式来执行任意系统命令.

最后 exp 如下, 写入so文件



修改 `LD_PRELOAD` 并转换图片格式, 最终执行系统变量 `EVIL_CMDLINE` 里的命令.



PS. 写入 so 文件的时候记得把 base64 里的 + 给编码, 不然会被当成空格.

还有一种思路是修改 PATH 环境变量, 修改为 `/tmp/xxx/`, 然后新建一个名为 `JxrEncApp` 之类的的恶意文件, `ImageMagick` 在 `delegete` 的时候将会调用这个, 从而执行命令.

不得不说题目质量是真的高, 膜 rr.

总结

除了这种 `LD_PRELOAD` 的方式之外, 还有其他一些骚操作, 可以看 l3m0n 师傅总结的.
在遇到这种困境时, 不防想想有没有什么其他办法能 bypass 掉当前的限制, 开辟新的方法.

上一篇：逻辑让我崩溃之验证码姿势分享 下一篇：0ctf2019 web writeup

1. 0 条回复
   - 动动手指，沙发就是你的了！