March 09, 2022

BROKEN ACCESS CONTROL TO POST-AUTH REMOTE CODE EXECUTION IN WEBMIN

NetbyteSEC Security Advisory - Broken Access Control To Post-Auth Remote Code Execution in Webmin

Title: Broken Access Control To Post-Auth Remote Code Execution in Webmin
Advisory ID: NBS-2022-0002
Product: Webmin
Vulnerable Version: <= 1.984
Fixed Version: 1.990
CVE ID: CVE-2022-0824, CVE-2022-0829
Homepage: https://www.webmin.com/

Date of Discovery: Feb 17th 2022
Author: Mohammad Faisal Sammio | NetbyteSEC

Vendor/product description:

Webmin is a web-based system administration tool for Unix-like servers, and services with over 1,000,000 installations worldwide. Using it, it is possible to configure operating system internals, such as users, disk quotas, services or configuration files, as well as modify, and control open-source apps, such as BIND DNS Server, Apache HTTP Server, PHP, MySQL, and many more.

Source: https://github.com/webmin/webmin

Vulnerabilities:

1) Improper Access Control to Post-Auth Remote Code Execution

CVE-ID: CVE-2022-0824
Risk: High, *Webmin marks as Critical
Vector: CVSS:3.0/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:L
Reference: https://nvd.nist.gov/vuln/detail/CVE-2022-0824
Description:

In Webmin v1.984, affecting the File Manager module, any authenticated low privilege user without access rights to the File Manager module could interact with second-level file manager functionalities such as download file from remote URL and change file permission (chmod). It is possible to achieve Remote Code Execution via a crafted .cgi file by chaining those functionalities in the file manager. This vulnerability is capable of modifying the OS file system and executing OS Command with root privileges.

Proof of Concept:

The underlying reason is that the CGI scripts in authentic-theme/extensions/file-manager did not properly check that the user has access to the File Manager module. Apparently, the default behaviour in Webmin for second-level access control for users who do not have access to a module is to grant maximal privileges because the Webmin believed that the primary check would prohibit them although it is not.

The first-level user access control functionalities such as save_file.cgi, upload.cgi, download.cgi and edit_file.cgi checking is properly done and protected so we could not abuse them to have write access to the server. Therefore, we have to discover the functionalities that are accessible by the second-level user access control to attain write access in order to place our CGI script. Fortunately, there is an endpoint that allows us to place our CGI script from a remote URL called http_download.cgi. Nevertheless, we could not execute that CGI script as the permission is *-rw-r--r-- (644)* [1] even we already have write access as shown in the code

snippet below where the Perl subroutine of *set_ownership_permissions(user, group, perms, file)* in line 48 shows that the 3rd argument which is perms argument is set to *undef* that refer to default umask value. When a user creates a file or directory in Linux or UNIX system, the permissions are set to the defaults. The root user's default umask is 022, which results in default directory permissions of 755 and default file permissions of 644.

File: \filemin\http_download.cgi - Line 48

```
[...]
    &set_ownership_permissions($st[4], $st[5], undef, $full);   //[1] 3rd argument, perms is set to undef
    @st = stat($cwd);
    print &text('http_done', &nice_size($st[7]),
          "<tt>".&html_escape($full)."</tt>"),"<p>\n";
    &ui_print_footer("index.cgi?path=".&urlize($path),
           $text{'previous_page'});
[...]
```

Further discovering allows us to identify an endpoint that could be used to modify the file permission called chmod.cgi which is self-explanatory. To achieve remote code execution as root privileges, it is sufficient to take advantage of two (2) second-level user access control accessible functionalities which are http_download.cgi and chmod.cgi endpoints as demonstrated in GitHub.

Reference: https://github.com/faisalfs10x/Webmin-CVE-2022-0824-revshell

2) Improper Authorization in Scheduled Cron Jobs Module

CVE-ID: CVE-2022-0829
Risk: Medium
Vector: CVSS:3.0/AV:N/AC:L/PR:L/UI:N/S:U/C:L/I:L/A:N
Reference: https://nvd.nist.gov/vuln/detail/CVE-2022-0829
Description:

The /cron/save_allow.cgi endpoint is accessible to any authenticated low privilege users resulting in controlling user access to cron jobs. They could allow and deny other users access to cron jobs affecting the Scheduled Cron Jobs module. This vulnerability is capable of modifying or restricting access to a system function outside the user's limits.

Proof of Concept:

File: \cron\save_allow.cgi - The save_allow.cgi endpoint is missing ACL permissions check. Hence, any low privilege user with access to the cron module could allow or restrict other users' access to the cron making the system user is unable to run scheduled tasks.

*** Developer fixed, adding ACL permission check [1] will fix the issue.

```
[...]
require './cron-lib.pl';
&ReadParse();

$access{'allow'} || &error($text{'allow_ecannot'});   //[1] add ACL permission check here

&lock_file($config{cron_allow_file});
&lock_file($config{cron_deny_file});
unlink($config{cron_allow_file});
unlink($config{cron_deny_file});
if ($in{mode} == 1) { &save_allowed(split(/\s+/, $in{'allow'})); }
elsif ($in{mode} == 2) { &save_denied(split(/\s+/, $in{'deny'})); }
&unlock_file($config{cron_allow_file});
&unlock_file($config{cron_deny_file});
&webmin_log("allow");
&redirect("");
[...]
```

Solution:

Update to the latest version 1.990. All systems with additional untrusted Webmin users should upgrade immediately.

Vendor Contact Timeline:
2022-02-17: Contact Webmin Security Contact (Jamie Cameron) via security[at]webmin.com
2022-02-21: Vendor response with acknowledgement and confirms security issue.
2022-03-03: Vendor releases security advisory and patches is available on version 1.990.
2022-03-06: Public release of security advisory.

NetByteSEC Sdn Bhd
==========

NetbyteSEC Sdn Bhd was incorporated under the Malaysian Companies Act 1965 in 2013.
NetbyteSEC is privately owned and is based in Nilai, Negeri Sembilan, Malaysia.
More information about NetbyteSEC Sdn Bhd can be found at:
https://www.netbytesec.com

COMMENTS

To leave a comment, click the button below to sign in with Google.

SIGN IN WITH GOOGLE

NetbyteSEC

Our Services

Contact Us

Trainings

About

This is a Cybersecurity blog from NetbyteSEC team aims to discuss threat research, tutorials, notes, advices, and opinions on current cybersecurity issues.