

[UNIX] Multiple up-imapproxy DoS Vulnerabilities

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2004-11/0057.html>

From: SecuriTeam (support_at_securiteam.com)

Date: 11/17/04

To: list@securiteam.com

Date: 17 Nov 2004 15:55:26 +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>

Multiple up-imapproxy DoS Vulnerabilities

SUMMARY

<<http://www.imapproxy.org/>> up-imapproxy is "an IMAP proxy that keeps connections open after client has logged out, and reuses it when client connects back. This is mostly useful for webmail-type clients".

There are various bugs in up-imapproxy that can crash it. Since up-imapproxy runs in a single process with each connection handled in a separate thread, any crash kills all the connections and stops listening for new ones. In 64bit systems it might be possible to make it leak data (mails, passwords, ..) from other connections to attacker's connection.

DETAILS

Vulnerable Systems:

* up-imapproxy version 1.2.2

up-imapproxy's literal sizes are stored in a signed long integer. This allows attacker to cause the server to crash by sending them when they were not "expected". Such is the case in `IMAP_Line_Read()`, which allows literals, but it's used in some places where literals weren't expected.

Any misuse will kill the proxy.

Securiteam: [UNIX] Multiple up-imapproxy DoS Vulnerabilities

One example is user/password in AUTHENTICATE LOGIN. In addition if literal is given to an unknown command, it's not properly handled. For example issuing the following command "x foo {5}" will kill the proxy.

Workaround:

Don't give direct IMAP access to up-imapproxy. It shouldn't be possible to exploit these bugs via webmails.

Fix:

Author seems to have gone away. Timo tried sending a few emails over a month ago with no reply. Its web site is currently broken too.

Timo does not really like saying "no fix available", so he wrote a patch which fixes the above problems. There might still be some problems left though. Note that Timo did only minimal testing with the patch.

```
diff -ru up-imapproxy-1.2.2/include/imapproxy.h
up-imapproxy-1.2.2-fixed/include/imapproxy.h
--- up-imapproxy-1.2.2/include/imapproxy.h 2004-07-23 16:17:24.000000000
+0300
+++ up-imapproxy-1.2.2-fixed/include/imapproxy.h 2004-11-07
18:51:00.000000000 +0200
@@ -206,7 +206,7 @@
    char ReadBuf[ BUFSIZE ]; /* Read Buffer */
    unsigned int BytesInReadBuffer; /* bytes left in read buffer */
    unsigned int ReadBytesProcessed; /* bytes already processed in read
buf */
- long LiteralBytesRemaining; /* num of bytes left to read as literal */
+ unsigned long LiteralBytesRemaining; /* num of bytes left to read as
literal */
    unsigned char NonSyncLiteral; /* rfc2088 alert flag */
    unsigned char MoreData; /* flag to tell caller "more data" */
    unsigned char TraceOn; /* trace this transaction? */
@@ -304,7 +304,7 @@
 */
extern int IMAP_Write( ICD_Struct *, const void *, int );
extern int IMAP_Read( ICD_Struct *, void *, int );
-extern int IMAP_Line_Read( ITD_Struct * );
+extern int IMAP_Line_Read( ITD_Struct *, int );
extern int IMAP_Literal_Read( ITD_Struct * );
extern void HandleRequest( int );
extern char *memtok( char *, char *, char ** );
diff -ru up-imapproxy-1.2.2/src/imapcommon.c
up-imapproxy-1.2.2-fixed/src/imapcommon.c
--- up-imapproxy-1.2.2/src/imapcommon.c 2004-07-23 16:17:25.000000000
+0300
+++ up-imapproxy-1.2.2-fixed/src/imapcommon.c 2004-11-07
18:54:05.000000000 +0200
@@ -428,7 +428,7 @@

    /* Read & throw away the banner line from the server */
```

Securiteam: [UNIX] Multiple up-imaproxy DoS Vulnerabilities

```
- if ( IMAP_Line_Read( &Server ) == -1 )
+ if ( IMAP_Line_Read( &Server, 0 ) == -1 )
  {
    syslog(LOG_INFO, "LOGIN: '%s' (%s:%d) failed: No banner line
received from IMAP server", Username, ClientAddr, sin_port );
    goto fail;
@@ -451,7 +451,7 @@
  /*
   * Read the server response
  */
- if ( ( rc = IMAP_Line_Read( &Server ) ) == -1 )
+ if ( ( rc = IMAP_Line_Read( &Server, 0 ) ) == -1 )
  {
    syslog(LOG_INFO, "STARTTLS failed: No response from IMAP
server after sending STARTTLS command" );
    goto fail;
@@ -555,7 +555,7 @@
  /*
   * the server response should be a go ahead
  */
- if ( ( rc = IMAP_Line_Read( &Server ) ) == -1 )
+ if ( ( rc = IMAP_Line_Read( &Server, 0 ) ) == -1 )
  {
    syslog(LOG_INFO, "LOGIN: '%s' (%s:%d) failed: Failed to
receive go-ahead from IMAP server after sending LOGIN command", Username,
ClientAddr, sin_port );
    goto fail;
@@ -611,7 +611,7 @@
  /*
   for ( ;; )
  {
- if ( ( rc = IMAP_Line_Read( &Server ) ) == -1 )
+ if ( ( rc = IMAP_Line_Read( &Server, 0 ) ) == -1 )
  {
    syslog(LOG_INFO, "LOGIN: '%s' (%s:%d) failed: No response from
IMAP server after sending LOGIN command", Username, ClientAddr, sin_port
);
    goto fail;
@@ -951,7 +951,8 @@
extern int IMAP_Literal_Read( ITD_Struct *ITD )
{
  char *fn = "IMAP_Literal_Read()";
- int Status, i, j;
+ int Status;
+ unsigned int i, j;
  struct pollfd fds[2];
  nfds_t nfds;
  int pollstatus;
@@ -1080,10 +1081,11 @@
  * process.
```

Securiteam: [UNIX] Multiple up-imapproxy DoS Vulnerabilities

```
*__
*/
-extern int IMAP_Line_Read( ITD_Struct *ITD )
+extern int IMAP_Line_Read( ITD_Struct *ITD, int useLiterals )
{
    char *CP;
- int Status, i, j;
+ int Status;
+ unsigned int i, j;
    char *fn = "IMAP_Line_Read()";
    char *EndOfBuffer;

@@ -1152,7 +1154,8 @@
    * string literal is coming next. How do we know?
    * If it is, the line will end with {bytecount}.
    */
- if ( ((CP - ITD->ReadBuf + 1) > 2) && ( *(CP - 2) == '}' ) )
+ if ( ((CP - ITD->ReadBuf + 1) > 2) && ( *(CP - 2) == '}' ) )
+ && useLiterals)
    {
        char *LiteralEnd;
        char *LiteralStart;
diff -ru up-imapproxy-1.2.2/src/main.c up-imapproxy-1.2.2-fixed/src/main.c
--- up-imapproxy-1.2.2/src/main.c 2004-07-23 16:17:25.000000000 +0300
+++ up-imapproxy-1.2.2-fixed/src/main.c 2004-11-07 18:52:41.000000000
+0200
@@ -931,7 +931,7 @@
    * The first thing we get back from the server should be the
    * banner string.
    */
- BytesRead = IMAP_Line_Read( &itd );
+ BytesRead = IMAP_Line_Read( &itd, 0 );
    if ( BytesRead == -1 )
    {
        syslog( LOG_ERR, "%s: Error reading banner line from server on
initial connection: %s -- Exiting.", fn, strerror( errno ) );
@@ -973,7 +973,7 @@
    * The second will be the OK response with the tag in it.
    */

- BytesRead = IMAP_Line_Read( &itd );
+ BytesRead = IMAP_Line_Read( &itd, 0 );
    if ( BytesRead == -1 )
    {
        syslog( LOG_ERR, "%s: Failed to read capability response from
server: %s -- exiting.", fn, strerror( errno ) );
@@ -986,7 +986,7 @@

    /* Now read the tagged response and make sure it's OK */
- BytesRead = IMAP_Line_Read( &itd );
```

Securiteam: [UNIX] Multiple up-imapproxy DoS Vulnerabilities

```
+ BytesRead = IMAP_Line_Read( &itd, 0 );
  if ( BytesRead == -1 )
  {
    syslog( LOG_ERR, "%s: Failed to read capability response from
server: %s -- exiting.", fn, strerror( errno ) );
@@ -1011,7 +1011,7 @@
  }

  /* read the final OK logout */
- BytesRead = IMAP_Line_Read( &itd );
+ BytesRead = IMAP_Line_Read( &itd, 0 );
  if ( BytesRead == -1 )
  {
    syslog( LOG_WARNING, "%s: IMAP_Line_Read() failed on LOGOUT --
Ignoring", fn );
diff -ru up-imapproxy-1.2.2/src/request.c
up-imapproxy-1.2.2-fixed/src/request.c
--- up-imapproxy-1.2.2/src/request.c 2004-07-23 16:17:26.000000000 +0300
+++ up-imapproxy-1.2.2-fixed/src/request.c 2004-11-07 19:05:09.000000000
+0200
@@ -433,6 +433,7 @@
  }

  strncpy( TraceUser, Username, sizeof TraceUser - 1 );
+ TraceUser[sizeof TraceUser - 1] = '\0';

  snprintf( SendBuf, BufLen, "%s OK Tracing enabled\r\n", Tag );
  if ( IMAP_Write( itd->conn, SendBuf, strlen( SendBuf ) ) == -1 )
@@ -611,7 +612,7 @@
  * The response from the client should be a base64 encoded version of
the
  * username.
  */
- BytesRead = IMAP_Line_Read( Client );
+ BytesRead = IMAP_Line_Read( Client, 0 );

  if ( BytesRead == -1 )
  {
@@ -654,7 +655,7 @@
    return( -1 );
  }

- BytesRead = IMAP_Line_Read( Client );
+ BytesRead = IMAP_Line_Read( Client, 0 );

  if ( BytesRead == -1 )
  {
@@ -1097,7 +1098,7 @@
  {
    do
    {
```

Securiteam: [UNIX] Multiple up-imapproxy DoS Vulnerabilities

```
- status = IMAP_Line_Read( Client );
+ status = IMAP_Line_Read( Client, 1 );

    if ( status == -1 )
    {
@@ -1152,7 +1153,7 @@
        if ( Server->LiteralBytesRemaining )
            break;

- status = IMAP_Line_Read( Server );
+ status = IMAP_Line_Read( Server, 1 );

        /*
         * If there's an error reading from the
server,
@@ -1266,7 +1267,7 @@
        if ( ! Client->NonSyncLiteral )
        {
            /* we have to wait for a go-ahead */
- status = IMAP_Line_Read( Server );
+ status = IMAP_Line_Read( Server, 0 );
            if ( Server->TraceOn )
            {
                sprintf( TraceBuf, sizeof TraceBuf - 1, "\n\n----->
C= %d %s SERVER: sd [%d]\n", time( 0 ), ( (TraceUser) ? TraceUser : "Null
username" ), Server->conn->sd );
@@ -1473,7 +1474,19 @@

        PollFailCount = 0;

- BytesRead = IMAP_Line_Read( &Client );
+ while ( Client.LiteralBytesRemaining )
+ {
+ BytesRead = IMAP_Literal_Read( &Client );
+
+ if ( BytesRead == -1 )
+ {
+ IMAPCount->CurrentClientConnections--;
+ close( Client.conn->sd );
+ return;
+ }
+ }
+
+ BytesRead = IMAP_Line_Read( &Client, 1 );

    if ( BytesRead == -1 )
    {
@@ -1530,6 +1543,7 @@
        * appropriate...
        */
        strncpy( S_Tag, Tag, MAXTAGLEN - 1 );
```

Securiteam: [UNIX] Multiple up-imapproxy DoS Vulnerabilities

```
+ S_Tag[MAXTAGLEN - 1] = '\0';
    if ( ! strcmp( (const char *)Command, "NOOP" ) )
    {
        cmd_noop( &Client, S_Tag );
@@ -1569,6 +1583,7 @@
        if ( Tag )
        {
            strncpy( S_Tag, Tag, MAXTAGLEN - 1 );
+ S_Tag[MAXTAGLEN - 1] = '\0';
            cmd_logout( &Client, S_Tag );
        }
    }
@@ -1641,7 +1656,8 @@
    }
    continue;
}
- strncpy( S_UserName, Username, sizeof S_UserName - 1 );
+ strncpy( S_UserName, Username, sizeof S_UserName - 1 );
+ S_UserName[sizeof S_UserName - 1] = '\0';

/*
 * Clients can send the password as a literal bytestream.
Check
@@ -1720,7 +1736,7 @@
    * IMAP_Literal_Read() right now since it works properly
    * otherwise.
    */
- rc = IMAP_Line_Read( &Client );
+ rc = IMAP_Line_Read( &Client, 1 );
    }
    else
    {
@@ -1748,6 +1764,7 @@

        *CP = '\0';
        strncpy( S_Password, Lasts, sizeof S_Password - 1 );
+ S_Password[sizeof S_Password - 1] = '\0';
    }

@@ -1779,6 +1796,7 @@
    if ( Tag )
    {
        strncpy( S_Tag, Tag, MAXTAGLEN - 1 );
+ S_Tag[MAXTAGLEN - 1] = '\0';
        cmd_logout( &Client, S_Tag );
    }
}

diff -ru up-imapproxy-1.2.2/src/select.c
up-imapproxy-1.2.2-fixed/src/select.c
--- up-imapproxy-1.2.2/src/select.c 2004-07-23 16:17:25.000000000 +0300
```

Securiteam: [UNIX] Multiple up-imapproxy DoS Vulnerabilities

```
+++ up-imapproxy-1.2.2-fixed/src/select.c 2004-11-07 18:56:01.000000000
+0200
@@ -356,7 +356,7 @@
    return( -1 );
}

- rc = IMAP_Line_Read( Server );
+ rc = IMAP_Line_Read( Server, 0 );

    if ( ( rc == -1 ) || ( rc == 0 ) )
    {
@@ -417,6 +417,7 @@
    ISC->ISCTime = time( 0 );

    strncpy( (char *)ISC->MailboxName, (const char *)MailboxName,
MAXMAILBOXNAME - 1 );
+ ISC->MailboxName[MAXMAILBOXNAME - 1] = '\0';

    return( 0 );
```

ADDITIONAL INFORMATION

The information has been provided by <<mailto:tss@iki.fi>> Timo Sirainen.

=====

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.