

up-imapproxy DoS vulnerabilities

Source: <http://www.derkeiler.com/Mailing-Lists/securityfocus/bugtraq/2004-11/0106.html>

From: Timo Sirainen (*tss_at_iki.fi*)

Date: 11/07/04

To: bugtraq@securityfocus.com

Date: Sun, 07 Nov 2004 20:12:18 +0200

Intro

up-imapproxy is an IMAP proxy which keeps connections open after client has logged out, and reuses it when client connects back. This is mostly useful for webmail-type clients.

Summary

There are various bugs in up-imapproxy which 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. However I don't think up-imapproxy actually works in any 64bit system so this is just a theoretical problem.

I'd also advise against using it's SELECT-cache feature (disabled by default), since its design is fundamentally broken. It may cause random problems with clients if the same mailbox is opened by multiple clients, or if the mailbox is modified behind up-imapproxy.

Details

Literal sizes were stored in a signed long integer. I wonder why since pretty much everything else was stored in unsigned integers.. Luckily this doesn't actually cause any buffer overflows in 32bit systems. With 64bit systems it would have allowed reading the buffer past it's boundaries (and thus possibly going into memory allocated by other connections).

IMAP_Line_Read() allows literals, but it's used in some places where literals aren't expected. Any misuse will kill the proxy. One example is

SecurityFocus Bugtraq: up-imapproxy DoS vulnerabilities

user/password in AUTHENTICATE LOGIN.

If literal is given to unknown command, it's not properly handled. For example next command after "x foo {5}" kills the proxy.

strncpy() is assumed to NUL-terminate the strings. I didn't see this causing any real problems though.

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. I tried sending a few emails over a month
ago with no reply. Its web site is currently broken too.
I don't really like saying "no fix available", so I wrote a patch which
fixes the above problems. There might still be some problems left though.
Note that I 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 *mementok( 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 */

-   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",
        goto fail;
@@ -451,7 +451,7 @@
    /*
     * Read the server response
     */
```

SecurityFocus Bugtraq: up-imaproxy DoS vulnerabilities

```
-     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");
        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");
        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 STARTTLS");
        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 @@
        *
        *--
        */
-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-imaproxy-1.2.2/src/main.c up-imaproxy-1.2.2-fixed/src/main.c
--- up-imaproxy-1.2.2/src/main.c      2004-07-23 16:17:25.000000000 +0300
+++ up-imaproxy-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
```

SecurityFocus Bugtraq: up-imapproxy DoS vulnerabilities

```
* 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 --
@@ -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.", f
@@ -986,7 +986,7 @@

    /* Now read the tagged response and make sure it's OK */
- 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
@@ -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 )
```

SecurityFocus Bugtraq: up-imapproxy DoS vulnerabilities

```

    {
@@ -1097,7 +1098,7 @@
    {
        do
        {
-           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 )
                    {
                        snprintf( TraceBuf, sizeof TraceBuf - 1, "\n\n-----> C= %d %s SERVER: sd [%d]
@@ -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 );
+           S_Tag[MAXTAGLEN - 1] = '\0';
                if ( ! strcasecmp( (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';
                    }
                }
            }
        }
    }
}

```

SecurityFocus Bugtraq: up-imapproxy DoS vulnerabilities

```
        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
+++ 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 );
```

SecurityFocus Bugtraq: up-imapproxy DoS vulnerabilities

- application/pgp-signature attachment: This is a digitally signed message part