

CRLF Injection

Source: <http://www.derkeiler.com/Mailing-Lists/securityfocus/bugtraq/2002-05/0077.html>

From: Ulf Harnhammar (ulfh@update.uu.se)

Date: 05/07/02

Date: Tue, 7 May 2002 00:12:10 +0200 (CEST)

From: Ulf Harnhammar <ulfh@update.uu.se>

To: bugtraq@securityfocus.com

CRLF Injection
by Ulf Harnhammar

"They crowded up to Lenin with their noses worn off / A handshake is worthy
if it's all that you've got"
— R.E.M., "Harborcoat"

Carriage Return (CR, ASCII code 13) and Line Feed (LF, ASCII code 10) are
two commonly used non-printing ASCII characters. Like many other useful
concepts in the world of computing, those characters and the treatment of
them have got security implications, particularly when they occur in
untrusted input.

LOG FILES

Log files consist of log entries, separated by LF, CRLF or just CR. If the
CR and LF characters are not removed from the input that is used to create
each log entry, we can make the log file change line and then create an
additional fake log entry.

As an example, let's say that we have a log file with a date field, a user
field and a comment field, like this:

```
2002-04-30 hans One log entry
2002-05-01 ulf Another log entry
```

If data is stored in that log file by the Perl statement:

```
print LOG "$date $user $comment\n";
```

and the \$comment variable is not checked for CR and LF characters, we can
add this additional fake entry:

```
2002-05-01 root This is serious!
```

by giving \$comment the value "Another log entry\n2002-05-01 root This is

SecurityFocus Bugtraq: CRLF Injection

serious!". At the same time that the "Another log entry" entry above is created, the program also creates the fake log entry.

INTERNET PROTOCOLS

Many network protocols used on the Internet define that a client should send a CRLF combination after each command that it sends to a server. If the CR and LF characters are not removed from the input that is used to put together the commands, we can send off several commands at the same time, where all but the first one are made up by us and just passed on.

The POP3 protocol uses the commands "RETR x" to retrieve messages and "DELE x" to delete them. If the client program constructs the command to retrieve a message with a "RETR \$msg\015\012" string, and the \$msg string is not checked for CR and LF characters, we can read message 1 while deleting message 2 by giving \$msg the value "1\015\012DELE 2". This will send the commands:

```
RETR 1
DELE 2
```

to the server.

The NNTP protocol uses the commands "ARTICLE x" to retrieve messages and "POST" to post them. If the client program constructs the command to retrieve a message with an "ARTICLE \$id\015\012" string, and the \$id string is not checked for CR and LF characters, we can read one message while silently posting another one.

Both of these cases have the following structure: "<KEY><SEP><VAL><NL>", where <KEY> might be "DELE", <SEP> is " ", <VAL> might be "1", and <NL> is CRLF. If the <VAL> field comes from user input and is allowed to contain the types of data found in the <KEY>, <SEP> and <NL> fields, the flaw exists.

MAIL, NEWS AND WEB HEADERS

E-mail headers, news headers and HTTP headers all have the structure "Key: Value", where each line is separated by the CRLF combination.

HTTP defines a "Location:" header for redirecting to another URL, and a "Set-Cookie:" header to set cookies. By embedding CR and LF characters in user input, web scripts can be fooled into setting a cookie from their own web server while redirecting to another site. If the script constructs the redirect with a "Location: \$url\015\012" string, and the \$url string is not checked for CR and LF characters, we can redirect to a site while setting a cookie by giving \$url the value "<http://www.kuro5hin.org/>\015\012Set-Cookie: evil=natas". If the cookie contains important data and one user can save URL's that another user will be redirected to, this can be serious.

Using the same technique, an e-mail system can reveal identities of people who were meant to be anonymous. Let's say that we have a system where users can

SecurityFocus Bugtraq: CRLF Injection

send e-mails to other users, but where the recipients' real e-mail addresses are hidden. If we are allowed to give the value of one mail header ourselves, like the "Subject:" header, and it is not checked for CR and LF characters, we can include a CRLF combination and then a "Bcc:" field with our own e-mail address in that "Subject:" header. At the same time that the message is sent to the recipient, it will also be silently sent to us, thus revealing the recipient's identity.

These cases also have the structure "<KEY><SEP><VAL><NL>" as described above. <KEY> might be "Subject", <SEP> is ":", <VAL> might be "CRLF Injection", and <NL> is CRLF.

SEVERITY

In some types of programs, this flaw may be a fatal blow to the program's security. In other cases, this may just be a small bug with low priority. It all depends on whether this flaw allows the user to do something he or she should not be able to do.

// Ulf Harnhammar
ulfh@update.uu.se

- *Previous message:* [KJK::Hyperion: "Nearly undocumented NT security feature – the solution to executable attachments?"](#)
- *Messages sorted by:* [\[date \]](#) [\[thread \]](#) [\[subject \]](#) [\[author \]](#) [\[attachment \]](#)