

MWR InfoSecurity  
Advisory

Crystal Reports

28<sup>th</sup> November 2006

MWR  INFOSECURITY

## CONTENTS

<b>1</b>	<b>Crystal Reports: Weak Sessions Advisory.....</b>	<b>3</b>
1.1	Overview: .....	3
1.2	Impact:.....	3
1.3	Cause:.....	3
1.4	Solution:.....	4
<b>2</b>	<b>Detailed Vulnerability Description .....</b>	<b>5</b>
2.1	Technical Background.....	5

## 1 Crystal Reports: Weak Sessions Advisory

<b>Package Name:</b>	Crystal Reports
<b>Date of release:</b>	2006-11-28
<b>Affected Versions:</b>	Business Objects Crystal Enterprise 10.0 Business Objects Crystal Enterprise 9.0

**CVE Reference:** CVE-2006-4099

**NISCC Reference:** 564575/NISCC/CRYSTALRPRTS

**Bugtraq ID:** 21350

**Severity:** High

**Access Vector:** Network exploitable

**Access Complexity:** Low

**Authentication:** Not required to exploit

**Impact Type:** Provides unauthorized access, Allows partial confidentiality, integrity, and availability violation, Allows unauthorized disclosure of information, Allows disruption of service

**Vendor URL:** [www.businessobjects.com](http://www.businessobjects.com)

**Vendor Response:** A Cumulative Hot Fix was issued on 9th June 2006 by business objects which fixed the vulnerability in Crystal Enterprise 10.

Crystal Reports 9 is an end-of-life product and Business Objects recommend that users upgrade to version 10. The mitigation section above still applies.

Individual Cumulative Hot Fixes do not have numbers, but all hot fixes released after the above contain the fix for this vulnerability.

**Exploit Details Included:** Yes

**OWASP Designation:** "Enter description here"

**Web Application Language:** "Enter description here"

### 1.1 Overview:

Crystal Reports makes use of a cookie value called WCSID as a session identifier. This session identifier is not sufficiently random, nor does it contain enough entropy. It contains a fixed portion, a time-based portion and an incremental portion. These are all predictable. In addition, the session identifier is not tied to a user's IP address. This combination allows for an attacker to hijack any currently authenticated users' sessions from any location.

### 1.2 Impact:

- Allows a remote, unauthenticated attacker full access to authenticated users' accounts.
- Various information disclosure issues.

### 1.3 Cause:

- Session ID generation is time-based and incremental

#### 1.4 Solution:

Apply patches as released by Business Objects.

It is important that as well as applying patches, Good application session management practice is followed. This would include such practices as (but not limited to) those listed below:

- Issue of a new cookie should be issued once a user has successfully authenticated.
- The cookie should be tied to a specific IP address for the duration of the session to prevent it being replayed from an arbitrary location. This resolution could have implications for users accessing the application across the Internet from particular ISPs.
- Security best practice also dictates that the SECURE flag should be set on the cookies once HTTPS has been implemented for the application.

## 2 Detailed Vulnerability Description

### 2.1 Technical Background

MWR InfoSecurity analysed the data contained within the WCSID cookie to determine whether it was susceptible to compromise. Initially, a large number of cookie values were recovered using a simple HTTP query tool. Each request made to the logon page returned a new cookie value.

A sample of this set of cookies follows:

Timestamp	Session Cookie
Fr1 Apr 21 12:52:36 BST 2006	11456203571015-:6401
Fr1 Apr 21 12:52:38 BST 2006	11456203571018-:6401
Fr1 Apr 21 12:52:37 BST 2006	11456203581017-:6401
Fr1 Apr 21 12:52:37 BST 2006	11456203581018-:6401
Fr1 Apr 21 12:52:38 BST 2006	11456203591019-:6401
Fr1 Apr 21 12:52:38 BST 2006	11456203591020-:6401
Fr1 Apr 21 12:52:39 BST 2006	11456203611021-:6401
Fr1 Apr 21 12:52:40 BST 2006	11456203611022-:6401
Fr1 Apr 21 12:52:40 BST 2006	11456203621023-:6401
Fr1 Apr 21 12:52:41 BST 2006	11456203621024-:6401
Fr1 Apr 21 12:52:42 BST 2006	11456203631025-:6401
Fr1 Apr 21 12:52:42 BST 2006	11456203631026-:6401
Fr1 Apr 21 12:52:43 BST 2006	11456203641027-:6401
Fr1 Apr 21 12:52:43 BST 2006	11456203641028-:6401
Fr1 Apr 21 12:52:44 BST 2006	11456203651029-:6401
Fr1 Apr 21 12:52:45 BST 2006	11456203651030-:6401
Fr1 Apr 21 12:52:45 BST 2006	11456203671031-:6401

Figure 1: Sample of WCSID session cookies

These values were analysed with a session analysis tool and the following graph highlights the linear graph that was returned when the difference between various cookies versus time was plotted.

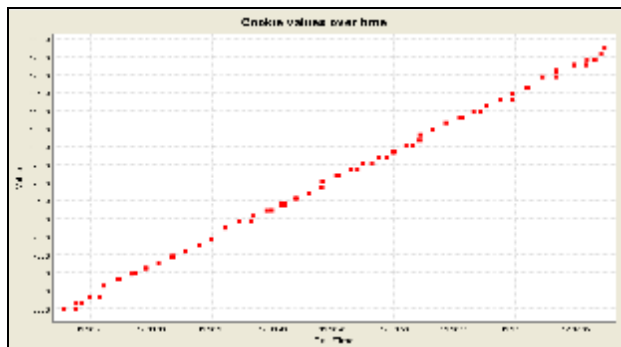


Figure 2: Cookie values have a linear relationship

As can be observed, the cookies returned contained a number of common elements and could therefore be broken down into a number of "pieces".

WCSID=11458808441941-1234567890:6401 (number changed to avoid identification)

This portion of the cookie remained static with every request and could therefore be easily predicted by an attacker. MWR InfoSecurity believes that this value is static for a particular installation and is linked to the license code required for Crystal Reports to access the back-end APS database. The value 6401 is the default TCP port number used by the APS application and reveals information about the communication path between the application server and database back-end.

WCSID=1145880844**1941**-1234567890:6401 (number changed to avoid identification)

The value in bold is an incremental counter and the value is increased with every new cookie that is issued. It is believed that this counter is present in the cookie to ensure that no two users will ever be granted the same cookie, even when accessing the page at the same time. This aspect of the cookie will be described in further detail later in this discussion. The counter values are in the range 0000 to 9999 and loops back round when the maximum value has been reached.

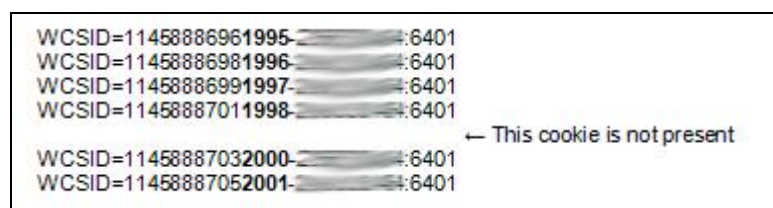
WCSID=**1145880844**1941-1234567890:6401 (number changed to avoid identification)

This portion of the cookie is also an incrementing counter; however, this value constantly increases by a small value every second. Therefore, the combination of these two incrementing counters ensure that each assigned session ID is unique. However, the complexity of the session identifier is not sufficient to protect the application from an attack.

A successful attack would require the attacker to know the two values of the counters at the time a legitimate user first accesses the site. The ability to perform such an attack is described in detail here.

Initially a script was constructed that requested the logon page on the server at intervals of roughly one second. The response to each query was filtered so that only the Cookie value was recorded and these were monitored until a recognisable gap appeared in the data-set.

The output included here demonstrates the attack being performed against a system located in the MWR InfoSecurity test lab.



```
WCSID=11458886961995-:6401
WCSID=11458886981996-:6401
WCSID=11458886991997-:6401
WCSID=11458887011998-:6401
← This cookie is not present
WCSID=11458887032000-:6401
WCSID=11458887052001-:6401
```

Figure 3: Inferring the value of a valid session cookie

As can be observed in the output, a legitimate user accessing the site has caused a gap to appear in the pattern, revealing a portion of their cookie to an attacker. In this instance, the legitimate user's cookie has that portion of the cookie set to 1999.

Therefore, the counter portion of the cookie that incremented with each request provides an attacker with information to identify when a user has accessed the site. In addition to providing the attacker with information about a valid session cookie, this counter also enables an attacker to perform analysis on the usage of the system.



This could be useful for a competitor or other party who is interested in this type of information.

The only portion of the cookie that is now required by an attacker is the portion that increments over time. However, the attacker's script has now set an upper and lower bound on the possible values and it is now simple to obtain this value. In this example the value lies between 1145888701 and 1145888703.

The discovery of value can be accomplished by using a script to request a page in the application that requires a valid cookie using every value between the maximum and minimum value. This attack must be completed after the user has been given sufficient time to logon to the application. Once this has occurred the user's session token will enable access to their account.

The impact is such that a remote unauthenticated attacker can gain full access to any account that is being accessed by users of the site.

A method of successfully hi-jacking user application sessions has been demonstrated by MWR InfoSecurity.

MWR InfoSecurity  
St. Clement House  
1-3 Alencon Link  
Basingstoke, RG21 7SB  
Tel: +44 (0)1256 300920  
Fax: +44 (0)1256 844083  
[mwrinfosecurity.com](http://mwrinfosecurity.com)