

MWR InfoSecurity Security
Advisory

IBM WebSphere MQ –
ziiVSendReceiveAgent
Memory Corruption
Vulnerability

4th March 2010



Contents

1	Detailed Vulnerability Description	4
1.1	Introduction	4
1.2	Technical Background.....	5
1.3	Exploit Information	6
1.4	Dependencies	6
2	Recommendations.....	7

ziiVSendReceiveAgent Memory Corruption Vulnerability

Package Name:	WebSphere MQ
Date:	2010-01-15
Affected Versions:	WebSphere 7.0.0.2 and 7.0.0.1 on Windows are confirmed to be vulnerable. Other versions and platforms may also be affected by this issue.

CVE Reference	CVE-2009-3160
Author	A. Plaskett
Severity	Medium Risk
Local/Remote	Remote
Vulnerability Class	Memory Corruption
Vendor URL	http://www-306.ibm.com/software/integration/wmq/
Vendor Response	A patch is available from the following URL: http://www-01.ibm.com/support/docview.wss?uid=swg24024153
Exploit Details Included	Yes (although no exploit code is provided).

Overview:

The WebSphere MQ service can be used to transfer messages between systems and applications. A memory corruption vulnerability was discovered that could allow an attacker to copy data outside the bounds of a memory page causing a denial of service condition and potentially code execution.

Impact:

The vulnerabilities discovered could lead to loss of service and disruption of legitimate usage. It is possible that the memory corruption vulnerability could also allow remote code execution; however, this has not been confirmed at this time.

Cause:

This vulnerability is caused by values in an MQ networking packet and which are user controlled being used as parameters to a function call with insufficient validation being performed on the size argument.

Interim Workaround:

Implementation of a security exit or SSL authentication would mitigate the risk from this vulnerability.

Solution:

The vendor supplied patches should be installed to resolve this issue. Links to the updated software can be found at the following location:
<http://www-01.ibm.com/support/docview.wss?uid=swg24024153>

1 Detailed Vulnerability Description

1.1 Introduction

WebSphere MQ is an Enterprise level application that can be used to provide a unified platform for messaging within an organisation. IBM describes their technology as follows:

“WebSphere MQ provides an award-winning messaging backbone for deploying your enterprise service bus (ESB) today as the connectivity layer of a service-orientated architecture (SOA).”

Source: <http://www-306.ibm.com/software/integration/wmq/>

Communication with MQ services can be achieved in a number of ways and the technology requires a feature rich API and extensions in order to integrate with an Enterprise environment.

The main component of a WebSphere MQ instance is the Queue Manager. This is a process that manages the message queues and provides interfaces (known as channels) to the applications wishing to communicate with them. By default, a Queue Manager will listen on a network interface for incoming connections and process the data accordingly. A Queue Manager will accept any type of MQ data and begin processing it before determining whether the packet is authorised or has been received at the correct point within the application’s “state machine”. The result of this fact is that a large amount of MQ code is exposed to unauthenticated users.

Consequently, any vulnerability in the code used to parse the data after it has been passed from the network socket can potentially be exploited by an attacker. After receiving data from the network socket the MQ application will process and parse the data in various ways. The exact nature of this parsing is not within the scope of this document; however, it is important to note that a large amount of this activity occurs before a connection to the Queue Manager is fully established. Once the parsing has been completed MQ will check whether the state machine is setup such that the packet belongs to a session that has been correctly established with the Queue Manager at the application level.

1.2 Technical Background

The “ziiVSendReceiveAgent” function is used by MQ to communicate using IPC through shared memory. A vulnerability exists in the “ziiVSendReceiveAgent” function when handling variable length strings (MQCHARV). A variable length string is composed of the following 4 byte fields (as documented at the following location:)
http://publib.boulder.ibm.com/infocenter/wmqv7/v7r0/topic/com.ibm.mq.csqzak.doc/fr22600_.htm

VSPtr	Pointer to the variable length string
VSOffset	Offset in bytes of the variable length string from the start of the structure that contains this MQCHARV structure
VSLength	The length in bytes of the variable length string addressed by the VSPtr or VSOffset field.
VSBufSize	The size in bytes of the buffer addressed by the VSPtr or VSOffset field.
VSCCSID	The character set identifier of the variable length string addressed by the VSPtr or VSOffset field.

The fields from the MQCHARV structure are handled by the following code:-

```
.text:508A947C          mov     ecx, [ecx+edi]
.text:508A947F          push   esi             ; size_t
.text:508A9480          add    ecx, eax
.text:508A9482          push   ecx             ; void *
.text:508A9483          push   ebp             ; void *
.text:508A9484          call  _memcpy
.text:508A9489          mov    eax, [esp+54h+var_24]
.text:508A948D          mov    ecx, [esp+54h+var_18]
.text:508A9491          add    esp, 0Ch
```

The ESI register is loaded with the VSLength value read from the data packet. This register is subsequently used to specify the number of bytes to be copied in the subsequent “memcpy” operation. Some input validation is performed on the length field before it is passed to this function as it is only possible to specify values that involve setting the two least significant bytes. Any attempt to alter the most significant bytes results in inconsistent results, although these have not been investigated any further at this point. The source of the “memcpy” operation as specified in the ECX register is also controllable as this is an offset into the data packet that is also read from the network data. Therefore, it is possible to control which data is copied into the destination and its length (within the restrictions noted above).

The validation performed on the VSLength value is sufficiently loose for it to be possible to specify lengths that are greater than the size of the destination and therefore force data to be written into locations that were not expected. In this instance, this is believed to be a shared memory segment that is potentially used by multiple MQ processes. The vulnerability therefore arises from the lack of bounds checking performed on the untrusted value provided in the MQCHARV VSLength field in the packet.

1.3 Exploit Information

In order to exploit this vulnerability an attacker could craft any MQ Packet which contains MQCHARV structures. Examples of packet types that support this data type are MQSUB and MQOPEN although others may exist. This vulnerability was exploited using an MQSUB packet and specifying the "SubUserName" MQCHARV.

MQCHARV data can be specified within the MQSUB API section of the packet and the following data will trigger an exception:

```
"\x00\x00\x00\x00" +  
"\x00\x00\x00\xff" + # Offset can control one byte  
"\x00\x00\xff\xff" + # Length of memcpy, can control two bytes.  
"\x00\x00\x00\x00" +  
"\x00\x00\x04\xb8"
```

This data causes an access violation due to the "memcpy" operation trying to read from outside the bounds of the memory page. However, if the source location was selected to be entirely within a valid memory page this operation would succeed and could allow the data to be written outside the intended destination.

As the size of the memory copy operation and the source offset can both be controlled by an attacker it may be possible to corrupt other memory data held in the page following and leverage this for code execution. It has so far not been possible to confirm the presence of any data that might be used for this purpose; however, the ability to perform a Denial of Service attack has been confirmed.

1.4 Dependencies

A connection must first be established to a Queue Manager and exploitation is therefore restricted to attackers with the ability to establish a network connection. The use of a security exit or SSL based authentication controls could mitigate the risk from this vulnerability, if they are implemented in an appropriate manner.

2 Recommendations

It is recommended that all users install the appropriate security patches released by the vendor in response to this issue. Links to the updated software can be discovered at the following location:

<http://www-01.ibm.com/support/docview.wss?uid=swg24024153>

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