

MWR InfoSecurity Security
Advisory

IBM WebSphere MQ
rriAcceptOAMUserAuth
Heap Overflow Vulnerability

2nd October 2009



Contents

1	Detailed Vulnerability Description	5
1.1	Introduction	5
1.2	Technical Background.....	5
1.3	Vulnerability Details.....	6
1.4	Exploit Information	7
1.5	Dependencies	7
2	Recommendations.....	8

IBM WebSphere MQ rriAcceptOAMUserAuth Heap Overflow

Package Name:	WebSphere MQ
Vendor Notified:	14-05-2009
Advisory Release Date:	02-10-2009
Affected Versions:	WebSphere MQ versions 7.0.0.1 and 6.0.0.0 and below on Windows are confirmed to be vulnerable. Patches are available for all previous versions and all operating systems supported by MQ version 6 and 7.

CVE Reference	CVE-2009-0896
Author	A. Plaskett
Severity	High Risk
Local/Remote	Remote
Vulnerability Class	Lack of input validation leading to Heap Buffer Overflow
Vendor Response	A patch is available from the following URLs: http://www-01.ibm.com/support/docview.wss?rs=171&uid=swg24023135 http://www-01.ibm.com/support/docview.wss?uid=swg21386826
Exploit Details Included	Yes (although no exploit code is provided)
Affected OS	Microsoft Windows is confirmed to be vulnerable It is expected that all OSs MQ supports are vulnerable due to the patches released to fix this issue.
References	http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2009-0896 http://www.securityfocus.com/bid/35170

Overview:

The WebSphere MQ service can be used to transfer messages between systems and applications. It has been identified that incorrect data validation is performed leading to a subsequent heap overflow vulnerability in the packet handling routines. This vulnerability is associated with the memory allocation code and can result in the overwriting of data on the heap. This vulnerability could be exploited remotely from an unauthenticated perspective in order to execute arbitrary code.

Impact:

The vulnerability could enable an attacker to remotely execute arbitrary code on the affected system with the privileges of the MQ process. An attacker could also use this vulnerability to create a Denial of Service condition for legitimate users.

Cause:

The vulnerability arises from insufficient input validation being performed on a number of values in the function rriAcceptOAMUserAuth which accepts data used to allocate a user specified amount of memory. This user specified amount of memory is then used to store an amount of user specified data. There is no checking performed to ensure that the amount of memory allocated is enough to store the data provided thus a Heap Overflow can occur.

Interim Workaround:

One method for mitigating the risk associated with this issue would be to use network filtering to restrict access to WebSphere MQ services to trusted IP addresses only.

Solution:

The vendor supplied patches should be installed to resolve this issue. Links to the updated software can be discovered at the following location: -

<http://www-01.ibm.com/support/docview.wss?uid=swg21386826>

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.

1.2 Technical Background

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 vulnerabilities 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.

The vulnerability is exposed when an unknown packet type with the segment type set to 0xA is received by the channel process ‘amqrmppa’. The ‘rriAcceptOAMUserAuth’ function will use values supplied within the packet as parameters to a number of memory allocation operations with no sanity checking performed.

1.3 Vulnerability Details

Multiple vulnerabilities were identified in the 'rriAcceptOAMUserAuth' function such that an attacker could affect the size of a number of memory allocations and the size of memory copied in the subsequent memcpy operations (shown below as the first heap memory allocation and first memcpy call).

The same series of vulnerable operations then occurs a second time; however, different values specified in the data packet are used (shown below as the second heap memory allocation and second memcpy call).

```

First heap memory allocation:
.text:4E864B68      mov     edx, [eax+24h]      ; value read from packet
.text:4E864B6B      lea    ecx, [edi+0A30h]
.text:4E864B71      push   ecx
.text:4E864B72      push   ebx
.text:4E864B73      push   edx                ; Size of memory to allocate
.text:4E864B74      push   1F6h
.text:4E864B79      push   14h
.text:4E864B7B      push   esi
.text:4E864B7C      call   _cccGetMem

First memcpy call:
.text:4E864C39      mov     ecx, [ecx+18h]     ; value read from packet
.text:4E864C3C      push   ecx                ; size_t
.text:4E864C3D      add    eax, 34h
.text:4E864C40      push   eax                ; void *
.text:4E864C41      push   edx                ; void *
.text:4E864C42      call   _memcpy

```

In the case of the first vulnerable memory allocation, the amount of heap memory to allocate using the cccGetMem wrapper function is read from the packet into the EDX register then pushed onto the stack as arguments for the function.

Subsequently, a further user specified value is read from the packet and used as the size argument of the memcpy call to copy data from the packet into the buffer allocated above.

If the size specified in the memcpy call is greater than the size of heap memory allocation (both of which are specified in the data packet) then both sets of operations can result in the overwriting of data on the heap with arbitrary data supplied in the packet.

```

Second heap memory allocation:
.text:4E864BC8      mov     eax, [eax+28h]     ; value read from packet
.text:4E864BCB      lea    edx, [edi+0A34h]
.text:4E864BD1      push   edx
.text:4E864BD2      push   ebx
.text:4E864BD3      push   eax                ; Size of memory to allocate
.text:4E864BD4      push   1F6h
.text:4E864BD9      push   14h
.text:4E864BDB      push   esi
.text:4E864BDC      call   _cccGetMem

Second memcpy call:
.text:4E864C69      push   edx                ; size_t
.text:4E864C6A      lea    edx, [ecx+eax+34h]
.text:4E864C6E      mov    eax, [edi+0A34h]
.text:4E864C74      push   edx                ; void *
.text:4E864C75      push   eax                ; void *
.text:4E864C76      call   _memcpy

```

Vulnerability Impact

An attacker capable of exploiting the vulnerability would be able to gain execution control within the process and thereby execute arbitrary code. The “amqrmppa” process, which handles data from network connections, runs with the privileges of the MQ user (this is dependent on the platform, for example, mqm on UNIX or MUSR_MQADMIN on Microsoft Windows). Therefore, any successful code execution would run with the privileges of this user account. Successful exploitation has been demonstrated against Windows 2000.

1.4 Exploit Information

The vulnerability exists within the ‘rriAcceptOAMUserAuth’ memory allocation function.

When an unknown MQ packet with the transmission segment type set to 0xA is received the rriAcceptOAMUserAuth function is called. The data specified in the packet is handled by the vulnerable function and used as described in Section 1.3. Therefore, a malicious packet can be created which causes the memory allocation to be smaller than the size specified in the subsequent memcpy operation.

The packet payload can be constructed to overwrite data structures on the heap after the allocated buffer. On the Windows 2000 platform it is possible to overwrite the subsequent heap header with the location of a known function (for example, PEB Lock) and the address of the shellcode to be executed. Once the overwritten header is used within the internal heap management it will cause the pointer to the PEB Lock function to be overwritten with the address of the shellcode.

This would mean that when a lock operation was next called the shellcode would be called instead and so control over execution would be gained. It should be noted that because of the flexibility over the size of memory controlled offered by this vulnerability it was possible to allocate one buffer to hold the shell code and the other (generally smaller) buffer to control the heap corruption.

MWR InfoSecurity have been able to construct a working exploit for Microsoft Windows platforms although there are no plans to release this code into the public domain at the present. Any decision to release such code in the future will be taken based on MWR InfoSecurity’s obligations to protect its customers and Critical National Infrastructure (CNI), whilst also allowing the security community to accurately assess the vulnerability of systems running the software.

1.5 Dependencies

This vulnerability has been tested on WebSphere MQ versions 7.0.0.1 and 6.0.0.0 on the Microsoft Windows platform. It should be noted that successful exploitation would be dependent on utilising a technique appropriate to the nature of the heap on the affected platform.

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=swg21386826>

Interim Workaround

Both network and host based traffic filtering controls could be implemented to protect access to the Queue Manager service on the affected hosts. This should be considered at both the packet filtering level and by the use of IPSec tunnels between hosts.

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