Checklist of Top MQ Security Outstanding Bugs/Issues/Gotchas

Change is the only constant

This presentation reflects…

- **My current opinions regarding WMQ security**
- **The product itself continues to evolve (even in PTFs)**
- **Attacks only get better with time**
- **This version of the presentation is based on WebSphere MQ v7.5 & v8.0**
- **This content will be revised over time so please be sure to check for the latest version at https://t-rob.net/links**
- **Your thoughts and ideas are welcome**

IMPORTANT NOTE

While at MQTC some of the CHLAUTH issues identified here were determined to be a probable bug, and some are associated with a behavior change introduced in a Fix Pack.

I am in the process of detailed testing of CHLAUTH behavior across multiple versions of MQ with ID/Password checking enabled in order to document the exact behavior and possibly submit PMRs to IBM.

With this in mind, please be aware that the information here on CHLAUTH is subject to significant change in the near future. I will post the updates as I have them to https://t-rob.net/ and the slides to https://t-rob.net/links.

Please be sure to check the web site for updates. If you are using CHLAUTH and having problems or wish to collaborate, please contact me!

Objectives

Finding it hard to secure your MQ estate?
MQ not behaving as documented?
Observed behavior not documented?

Cheer up, it's not you. Often, it's MQ, if that's any consolation. In this session we will cover MQ security bugs, defects and gotchas in three categories:

1) **Stuff that's broken by accident;**

2) **Stuff that's broken by design; and**

3) **Stuff that's nothing more than security theater.**

If we have time, we'll vote to pick the fan favorites.

Obligatory disclaimer

The characterization of things as being broken by accident, design, or amounting to nothing more than security theater is my opinion.

In particular, when I say “broken by design” it refers to something that is working as documented, meets its design goals, won’t be fixed under a PMR, but has a negative impact on the ability to secure the queue manager.

It does not mean that anyone at IBM intentionally designed for weak security.

The things that are security theater are on us, the customers. Sometimes we demand stuff in the name of security that is, shall we say, sub-optimal. Given sufficient demand, IBM has almost no choice but to deliver it, even if doing so displaces something that would provide more meaningful control.

Mostly though when security in MQ is broken, it’s by accident. I spend the least time on these here since they are fix candidates the moment they are acknowledged as bugs. I’ll call out a couple of the big ones though.

Agenda

1) **Stuff that's broken by accident;**
   - Local runmqsc does what?!?!
   - Remote PCF
   - After the fix local MQSC does what?!?!

2) **Stuff that's broken by design; and**
   - ID Context
   - CHLAUTH combining
   - Kdb conversion loses sth, crl, loses kdb in prior versions
   - Security xor event monitoring

3) **Stuff that's nothing more than security theater.**
   - REVDNS
   - CHKCLNT(NONE && REQUIRED)
   - Built for enumeration

Local runmqsc does what?!?!?

- Users with only +connect on the QMgr can issue REFRESH SECURITY commands.

When runmqsc was changed to manage CCDT files independently of the QMgr, that meant non-mqm users needed to run it. It ended up with the same problem as any other program that's been running with full admin and subsequently expected to run with restricted rights – security has never been tested. No surprise that some bits of it don't quite work.

This affects only locally-connected runmqsc sessions. Client-connected sessions behave like any other client.

Affects Versions:

- 8.0.0.0 – v8.0.0.4.
- 7.0, 7.1, 7.5 but I have not validated the specific Fix Pack in which it was addressed for each. Look for CVE-2015-7473.

Remote PCF

With only +connect on the QMgr plus normal access to Command and model queues, it was possible to display:

- **Channel status – DIS CHS(*)**
- **Queue status – DIS QS(*)**
- **Auth records – DIS AUTHREC(*)**
- **Connections – DIS CONN(*)**

```
Fix delivered in 8.0.0.4
Have yet to check other major versions.
```

Remote PCF

Command	Before fix	After fix	Auth required
dis qmgr	2035	2035	**
dis conn(*)	No error	2035	dsp on qmgr
dis conn(*) all	No error	2035	dsp on qmgr
dis qs(*)	No error	2035	dsp on queue
dis qs(*) all	No error	2035/No status	dsp on queue
dis chl(*)	2035	2035	**
dis chs(*)	No error	No error	**
dis sub(*)	2035	2035	**
dis sub(*) all	2035	2035	**
dis sbstatus(*)	2035	2035	**
dis sbstatus(*) all	2035	2035	**
dis topic(*)	2035	2035	**
dis topic(*) all	2035	2035	**
dis chlauth(*)	2035	2035	**
dis authinfo(*)	2035	2035	**
dis authinfo(*) all	2035	2035	**
dis authrec	No error	2035	dsp on qmgr

** - No change in the behavior after the fix.
Fix delivered in 8.0.0.4

Agenda

1) **Stuff that's broken by accident;**
   - Local runmqsc does what?!?!
   - Remote PCF
   - After the fix local MQSC does what?!?!

2) **Stuff that's broken by design; and**
   - ID Context
   - CHLAUTH combining
   - Kdb conversion loses sth, crl, loses kdb in prior versions
   - Security xor event monitoring

3) **Stuff that's nothing more than security theater.**
   - REVDNS
   - CHKCLNT(NONE && REQUIRED)
   - Built for enumeration

CHLAUTH doesn’t work like you think

```
SET CHLAUTH('SVRCONN.CHL') TYPE(USERMAP) +
      CLNTUSER('admusr2') CHCKCLNT(REQUIRED) +
      USERSRC(MAP) MCAUSER('mqm') ACTION(replace)
```

Whether you get admusr2 or mqm as the MCAUSER depends on the value of ADOPTCTX which is a QMgr-global setting.

The rule still applies when ADOPTCTX(YES) but the mapping is completely ignored. You can use the filtering features of SSL mapping, User mapping or address mapping rules, just don’t expect the mapping to actually occur.

End result is that the channel can have one MCAUSER, the mapping rule has another, and the channel status MCAUSER is matches neither.

CHLAUTH doesn’t work like you think

CHLAUTH Combining

SET CHLAUTH('SVRCONN.CHL') TYPE(ADDRESSMAP) DESCR('Block all access by default') ADDRESS('*') USERSRC(NOACCESS)

SET CHLAUTH('SVRCONN.CHL ') TYPE(SSLPEERMAP) SSLPEER('CN=*, OU=MQ Admin') USERSRC(CHANNEL)

SET CHLAUTH('SVRCONN.CHL') TYPE(USERMAP) CLNTUSER(‘t.rob') CHCKCLNT(REQUIRED) USERSRC(MAP) MCAUSER('mqm')

These rules were intended by the MQ Admin to be hierarchical:

1. **Match to certificates with “MQ Admin” as the OU,**
2. **Force password check**
3. **Map Admin users to mqm**

Only the SSL rule actually fires. Since the channel has MCAUSER(*nobody) the result is the connection is blocked.

Agenda

1) **Stuff that's broken by accident;**
   - Local runmqsc does what?!?!
   - Remote PCF
   - After the fix local MQSC does what?!?!

2) **Stuff that's broken by design; and**
   - ID Context
   - CHLAUTH combining
   - Kdb conversion loses sth, crl, loses kdb in prior versions
   - Security xor event monitoring

3) **Stuff that's nothing more than security theater.**
   - REVDNS
   - CHKCLNT(NONE && REQUIRED)
   - Built for enumeration

CHLAUTH doesn’t work like you think

SET CHLAUTH('SVRCONN.CHL') TYPE(ADDRESSMAP) DESCR('Block all access by default') ADDRESS('*') USERSRC(NOACCESS)

SET CHLAUTH('SVRCONN.CHL ') TYPE(SSLPEERMAP) SSLPEER('CN=t.rob, OU=MQ Admin') USERSRC(MAP) MCAUSER(‘mqm’)

~~**SET CHLAUTH('SVRCONN.CHL') TYPE(USERMAP) CLNTUSER(‘t.rob') CHCKCLNT(REQUIRED) USERSRC(MAP) MCAUSER('mqm')**~~

We got rid of the rule that doesn’t fire but we now need an SSL rule for each certificate.

If CHKCLNT(REQUIRED) and ADOPTCTX(YES) the MCAUSER doesn’t end up being mqm.

Agenda

1) **Stuff that's broken by accident;**
   - Local runmqsc does what?!?!
   - Remote PCF
   - After the fix local MQSC does what?!?!

2) **Stuff that's broken by design; and**
   - ID Context
   - CHLAUTH combining
   - Kdb conversion loses sth, crl, loses kdb in prior versions
   - Security xor event monitoring

3) **Stuff that's nothing more than security theater.**
   - REVDNS
   - CHKCLNT(NONE && REQUIRED)
   - Built for enumeration

Where's my KDB?

GSKit is (usually) very helpful. Type in a partial command, it gives you the syntax for the remainder.

Unless you try to convert a KDB. D'oh!

In prior versions, GSKit made the existing files into temp files before parsing the whole command. If it failed mid-stream, it deleted the temp files leaving you with no KDB!

The "fixed" version makes the command syntactically correct. Now if you leave off the target DB and type, it just converts from KDB to KDB. With a slight wrinkle.

If you do not specify -stashed it will delete the stash file! If you do specify it preserves the stash file. Obviously you need to know the password if you leave the -stashed parm off but if you fail to notice the lack of a stash file and do not recreate it that might be a problem later.

Security xor event monitoring

We all need more granular security than “Admins and Everyone Else.”

We’d *like* better monitoring but it has to be meaningful and not generate alerts that routinely need to be ignored.

- **DIS Q(*) generates authorization event for each queue to which the requestor is not authorized.**
- **CMDEV(NODISPLAY) doesn’t help.**

Agenda

1) **Stuff that's broken by accident;**
   - Local runmqsc does what?!?!
   - Remote PCF
   - After the fix local MQSC does what?!?!

2) **Stuff that's broken by design; and**
   - ID Context
   - CHLAUTH combining
   - Kdb conversion loses sth, crl, loses kdb in prior versions
   - Security xor event monitoring

3) **Stuff that's nothing more than security theater.**
   - REVDNS
   - CHKCLNT(NONE && REQUIRED)
   - Built for enumeration

Reverse DNS

We authenticate SVRCONN channels on the principle that the QMgr should not blindly accept the ID that is presented to it as being authentic.

But it should blindly accept the DNS name presented to it as being authentic?

- **DNS spoofing**
- **Client endpoint impersonation**
- **MAC impersonation**
- **Man-in-the-middle**
- **Aggregate identities (VPN exit nodes, gateways, load balancers, etc.)**
- **Bots**

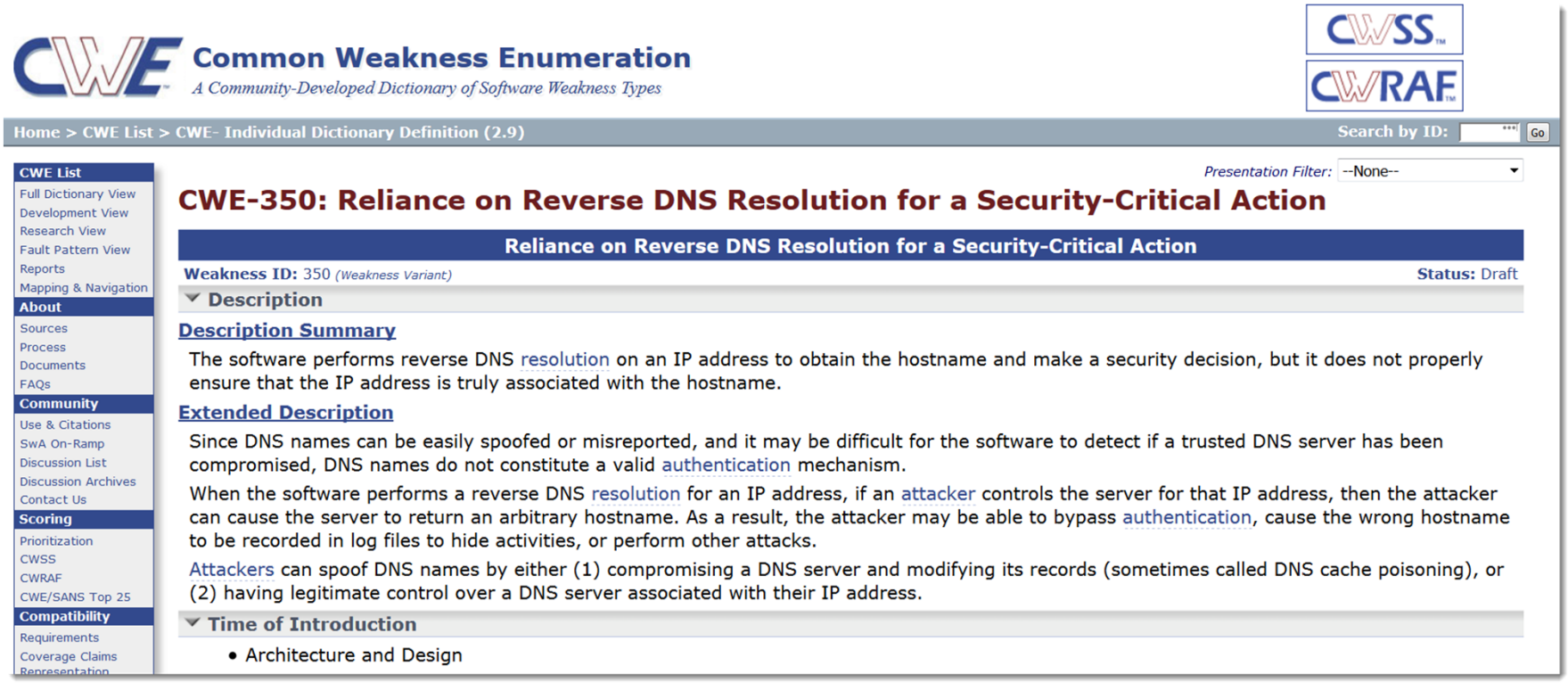
Reverse DNS

Oh yeah – and sometimes it doesn't work.

- **DNS resolution across the firewall can hang.**
- **Connection may originate from multiple interfaces.**
- **Forward resolution exists but no PTR for reverse resolution.**
- **Multiple PTR for same name but different IP addresses.**
- **In original incarnation would perform lookups on all IP addresses presented whether authentication was required or not. (May since have been fixed.)**

Per IBM PMR, function of rDNS checking with some RFC-compliant configurations (such as multiple PTR for an IP) is non-deterministic. To employ something with non-deterministic behavior as a security control is a textbook instance of Security Theater.

Don’t take my word for it



Common Weakness Enumeration
A Community-Developed Dictionary of Software Weakness Types

CWSS
CWRAF

Home > CWE List > CWE- Individual Dictionary Definition (2.9)

Search by ID: Go

CWE List
Full Dictionary View
Development View
Research View
Fault Pattern View
Reports
Mapping & Navigation
About
Sources
Process
Documents
FAQs
Community
Use & Citations
SwA On-Ramp
Discussion List
Discussion Archives
Contact Us
Scoring
Prioritization
CWSS
CWRAF
CWE/SANS Top 25
Compatibility
Requirements
Coverage Claims
Representation

Presentation Filter: --None--

CWE-350: Reliance on Reverse DNS Resolution for a Security-Critical Action

Reliance on Reverse DNS Resolution for a Security-Critical Action

Weakness ID: 350 (Weakness Variant) Status: Draft

Description

Description Summary

The software performs reverse DNS resolution on an IP address to obtain the hostname and make a security decision, but it does not properly ensure that the IP address is truly associated with the hostname.

Extended Description

Since DNS names can be easily spoofed or misreported, and it may be difficult for the software to detect if a trusted DNS server has been compromised, DNS names do not constitute a valid authentication mechanism.

When the software performs a reverse DNS resolution for an IP address, if an attacker controls the server for that IP address, then the attacker can cause the server to return an arbitrary hostname. As a result, the attacker may be able to bypass authentication, cause the wrong hostname to be recorded in log files to hide activities, or perform other attacks.

Attackers can spoof DNS names by either (1) compromising a DNS server and modifying its records (sometimes called DNS cache poisoning), or (2) having legitimate control over a DNS server associated with their IP address.

Time of Introduction

- Architecture and Design

Reverse DNS as a security control is such a glaring anti-pattern that it has its own CWE.

Example CVEs and mitigations

Observed Examples

Reference	Description
CVE-2001-1488	Does not do double-reverse lookup to prevent DNS spoofing.
CVE-2001-1500	Does not verify reverse-resolved hostnames in DNS.
CVE-2000-1221	Authentication bypass using spoofed reverse-resolved DNS hostnames.
CVE-2002-0804	Authentication bypass using spoofed reverse-resolved DNS hostnames.
CVE-2001-1155	Filter does not properly check the result of a reverse DNS lookup, which could allow remote attackers to bypass intended access restrictions via DNS spoofing.
CVE-2004-0892	Reverse DNS lookup used to spoof trusted content in intermediary.
CVE-2003-0981	Product records the reverse DNS name of a visitor in the logs, allowing spoofing and resultant XSS.

Potential Mitigations

Phase: Architecture and Design

Use other means of identity verification that cannot be simply spoofed. Possibilities include a username/password or certificate.

ID/Password or certificate authentication are recommended authentication alternatives to reverse DNS. If only MQ could perform password or cert authentication, we'd have meaningful security. Oh wait… it does. We do.

CHKCLNT(NONE && REQUIRED)

No way to have one channel have CHCKCLNT(NONE) and another have CHCKCLNT(REQUIRED).

With CHCKCLNT(OPTIONAL) if MQ finds a username populated in the MQCSP (in this case password is expected to be right so even a null value is wrong) or for compatibility with Java API/JMS API if MQ finds the MQCD RemoteUserIdentifier and RemotePassword fields, MQ will authenticate those values.

The OPTIONAL only applies if MQ finds no MQCSP and no MQCD RemotePassword.

The issue with this is that you have no way to have MQ require connauth on some channels and have other channels where you want MQ to ignore the username and password that is presented.

Scenario: ID and password authentication of interactive users and certificate authentication of client applications.

Agenda

1) **Stuff that's broken by accident;**
   - Local runmqsc does what?!?!
   - Remote PCF
   - After the fix local MQSC does what?!?!

2) **Stuff that's broken by design; and**
   - ID Context
   - CHLAUTH combining
   - Kdb conversion loses sth, crl, loses kdb in prior versions
   - Security xor event monitoring

3) **Stuff that's nothing more than security theater.**
   - REVDNS
   - CHKCLNT(NONE && REQUIRED)
   - Built for enumeration

A user by any other name

No way to adopt the User ID from password authentication on some channels but not others.

Scenario for ID/Password authentication:

- **Use the authenticated user ID for non-admin users, app service accounts.**
- **Use combination of certificates and password to strongly authenticate administrators, then map the user ID to mqm.**
  This is because the mqm group is empty and these users are not in it. If you are wondering why not add the admins to the mqm group, come see me after the session.

Possible mitigations:

- **Ideally, downgrade to CHKCLNT(NONE) on a per-channel or per-CHLAUTH.**
- **AUTHINFO has CHKCLNT(NONE) and uplift per-channel or per CHLAUTH.**

A user by any other name – Pt II

No way to filter on certificate *and* use ID & password auth.

Only one CHLAUTH rule fires so choices are:

- **User map but no SSL map**
- **SSL Map but no user map**
- **ADOPTCTX(YES) in which case the mapping portion of any CHLAUTH rule never fires.**

Remember REVDNS? No chance the mapping portion of any rule specifying a DNS entry will fire if ADOPTCTX(YES). On the other hand if ADOPTCTX(NO) starts with the UserID from the channel which is the MCAUSER from the channel definition which (hopefully!) should be something like *NOACCESS.

Agenda

1) **Stuff that's broken by accident;**
   - Local runmqsc does what?!?!
   - Remote PCF
   - After the fix local MQSC does what?!?!

2) **Stuff that's broken by design; and**
   - ID Context
   - CHLAUTH combining
   - Kdb conversion loses sth, crl, loses kdb in prior versions
   - Security xor event monitoring

3) **Stuff that's nothing more than security theater.**
   - REVDNS
   - CHKCLNT(NONE && REQUIRED)
   - Built for enumeration

MQ is built for resource enumeration

Whenever I pitch Defense in Depth or things that prevent a breach from spreading (blast radius containment) we end up discussing exploits for which the attacker needs to know names of objects, users, queue managers, etc.

“That’s not a big risk because an attacker would have to know ‘x’ to exploit the exposure. If MQ is properly authorized they can’t know that.”

a) **If the QMgr is breached, they know that. Design security to withstand or at least contain a breach.**
b) **If structured data is used for object names (i.e. branch ID) they can derive that.**
c) **If they are a legitimate user, they can see much of that and MQ shows far more than would be prudent from a security perspective.**

MQ is built for resource enumeration

IBM MQ classes for JMS applications need +connect and +inq authority to the queue manager. It needs to get the QMgr name, DLQ name, etc. Unfortunately, that *also* means any JMS app can see…

- **Inta-group queuing ID, QSG name**
- **Installation details**
  - MQCA_INSTALLATION_DESC
  - MQCA_INSTALLATION_NAME
  - MQCA_INSTALLATION_PATH
- **MQCA_REPOSITORY_NAME, MQCA_REPOSITORY_NAMELIST**
- **Max Channels, Max Handles, Max Uncommitted Messages**
- **All event settings**
- **Whether queued Pub/Sub is enabled**
- **Whether Advanced Message Security is in use**
- **AdoptNewMCA values**

http://www.ibm.com/support/knowledgecenter/en/SSFKSJ_8.0.0/com.ibm.mq.dev.doc/q031710_.htm
http://www.ibm.com/support/knowledgecenter/en/SSFKSJ_8.0.0/com.ibm.mq.ref.dev.doc/q101840_.htm

MQ is built for resource enumeration

The client app may not have access to the local command queue but if any XMitQ is accessible, chances are good the adjacent command queue is too.

- **MQOD.ResolvedQName, MQOD.ResolvedQMgrName**
- **Names with structured data? 2035==exists, 2085==not-exists.**
- **Once a route to an adjacent QMgr is found…**
  - Create a QRemote pointing to the local QMgr's command queue.
  - Send DIS QL(*), DIS QR(*), DIS QA(*), DIS CHA(*), all of which work.

Acknowledgements

This presentation would not be possible without the dedicated efforts of many people in the community and in IBM who document, validate, report, track, test and share security issues. The folks named below have all contributed significantly to this presentation.

- **From the community: Josh McIver, Neil Casey, Peter Potkay, Morag Hughson, Paul Clarke**
- **From IBM: Jeff Lowrey, Andrew Hickson, Rob Parker, David Ware, Pete Murphy, Mark Taylor, Bill Oppenheimer, David Chrighton, Trevor Dolby, Barry Spiers**

Questions & Answers