Your Q is my Q

Message Queue Security

G. Geshev

NoSuchCon 2014
Paris, France

Introduction

Georgi Geshev

- Security Researcher at MWR Labs
- Research Interests
  - Vulnerability Development
  - IPv6 Network Reconnaissance
  - Message Queues

Agenda

- MQ Concepts
- Attack Surface
- Case Studies
- Attack Scenarios
- Common Issues
- MQ Hardening

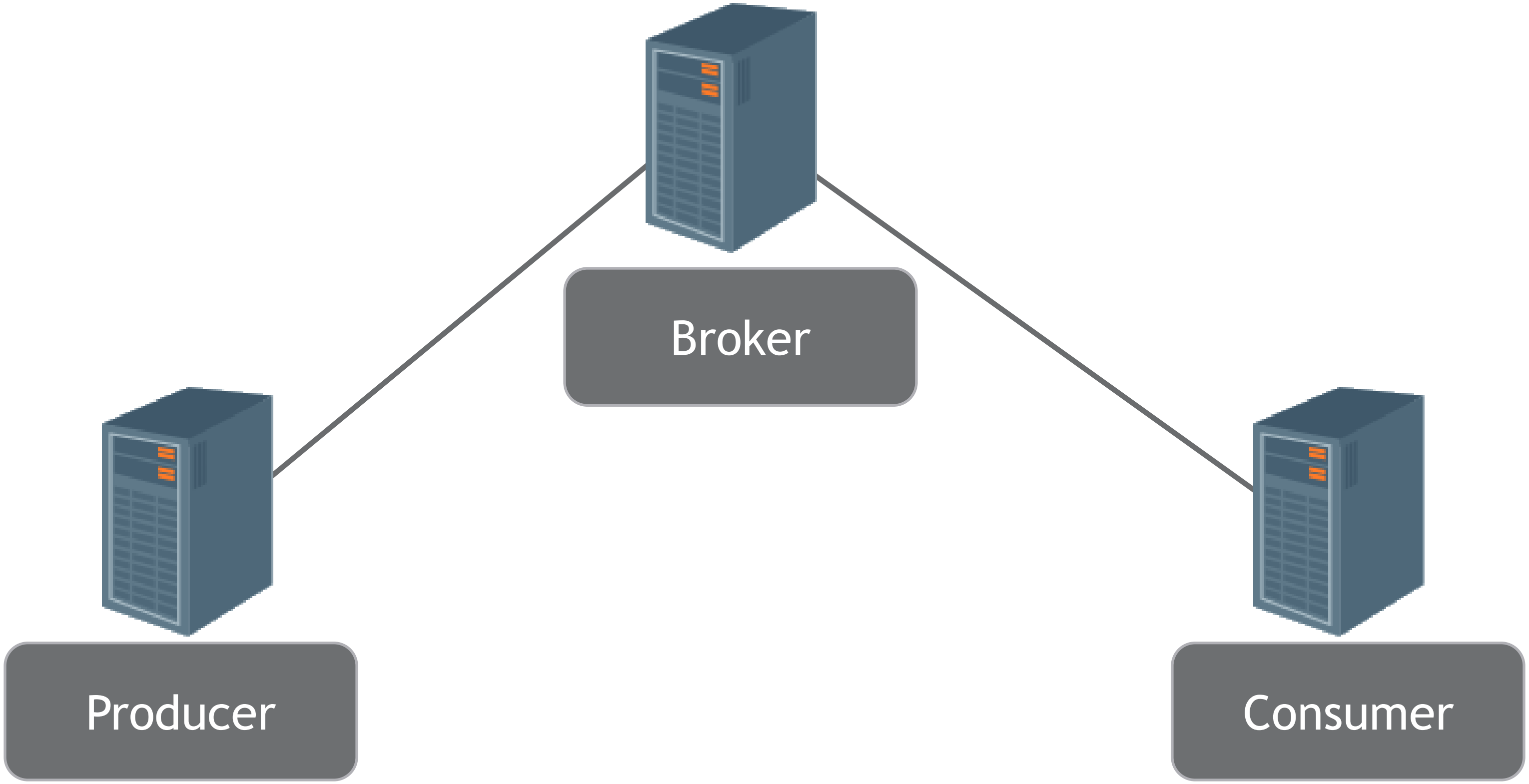
Disclaimer

- This is **not** a talk on new classes of bugs, i.e. none of the vulnerabilities are MQ specific.
- This **is** a talk on problems found to be common across some popular MQ implementations.

MQ Concepts

- Message-oriented Middleware (MOM)
  - Asynchronous Message Exchange
  - Decoupling
    - Space, Time and Synchronization Decoupling
  - Publish & Subscribe
    - Publishers Create Messages
    - Subscribers Consume Messages
    - Topic, Content and Type Based Subscriptions

MQ Concepts

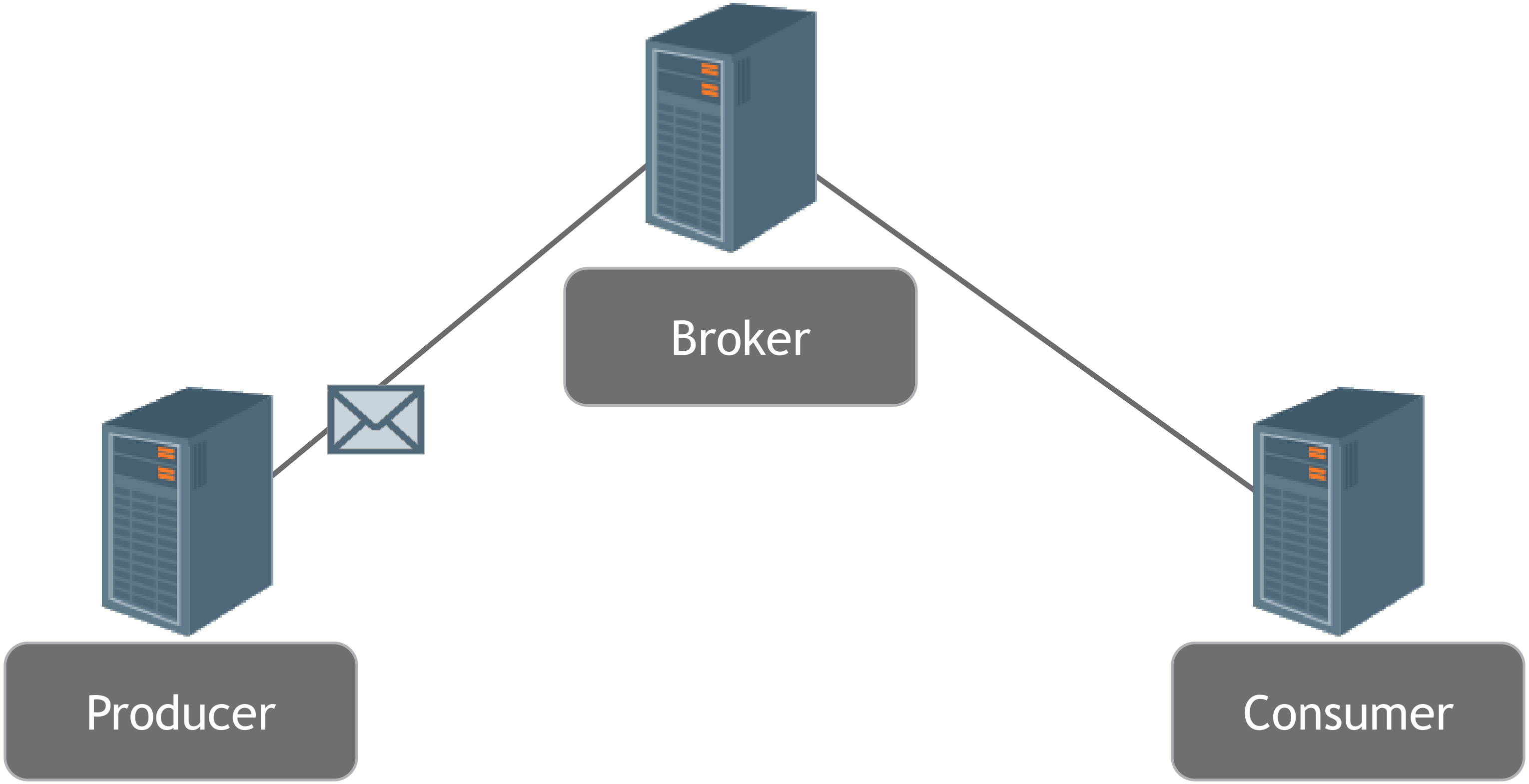


Broker

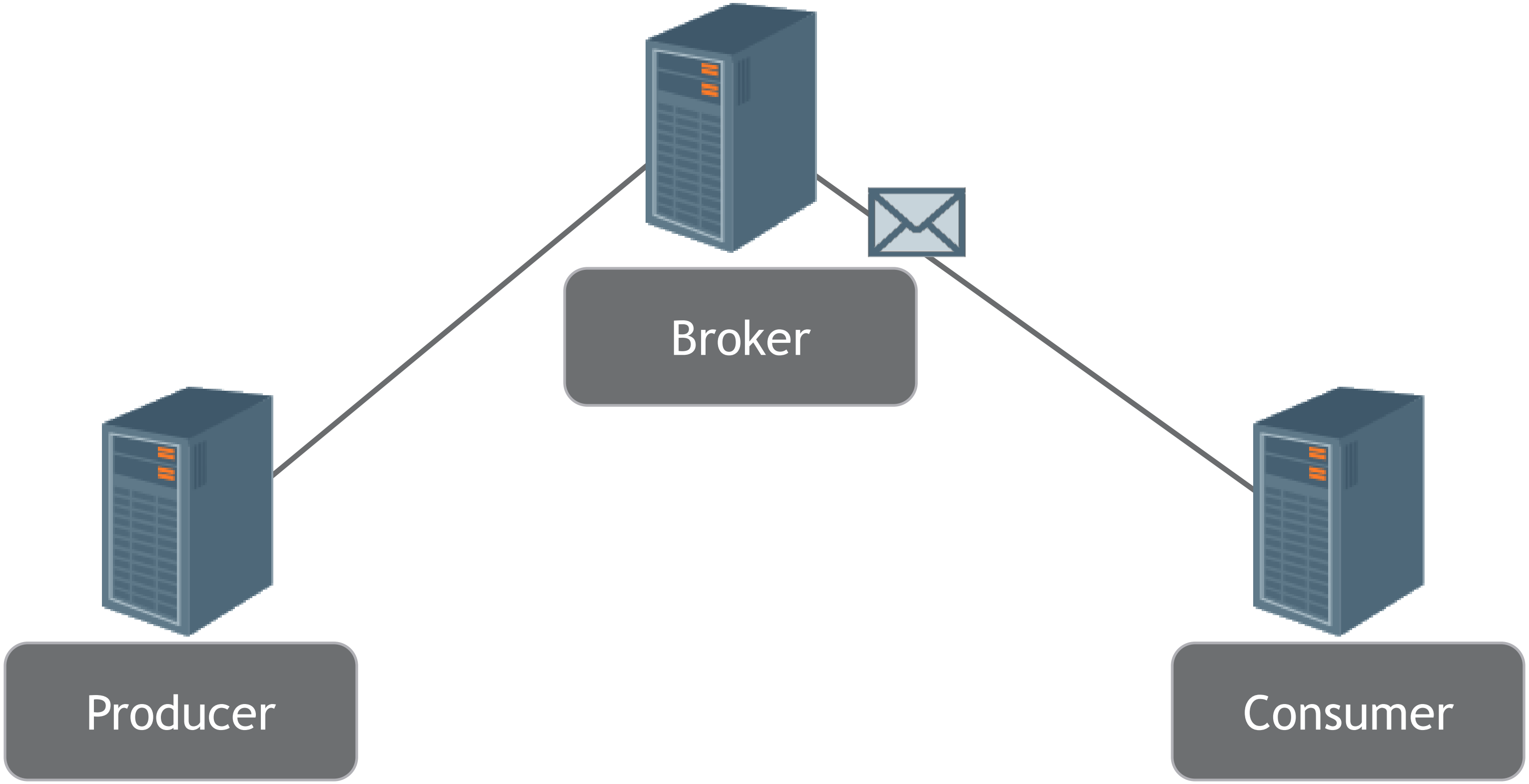
Producer

Consumer

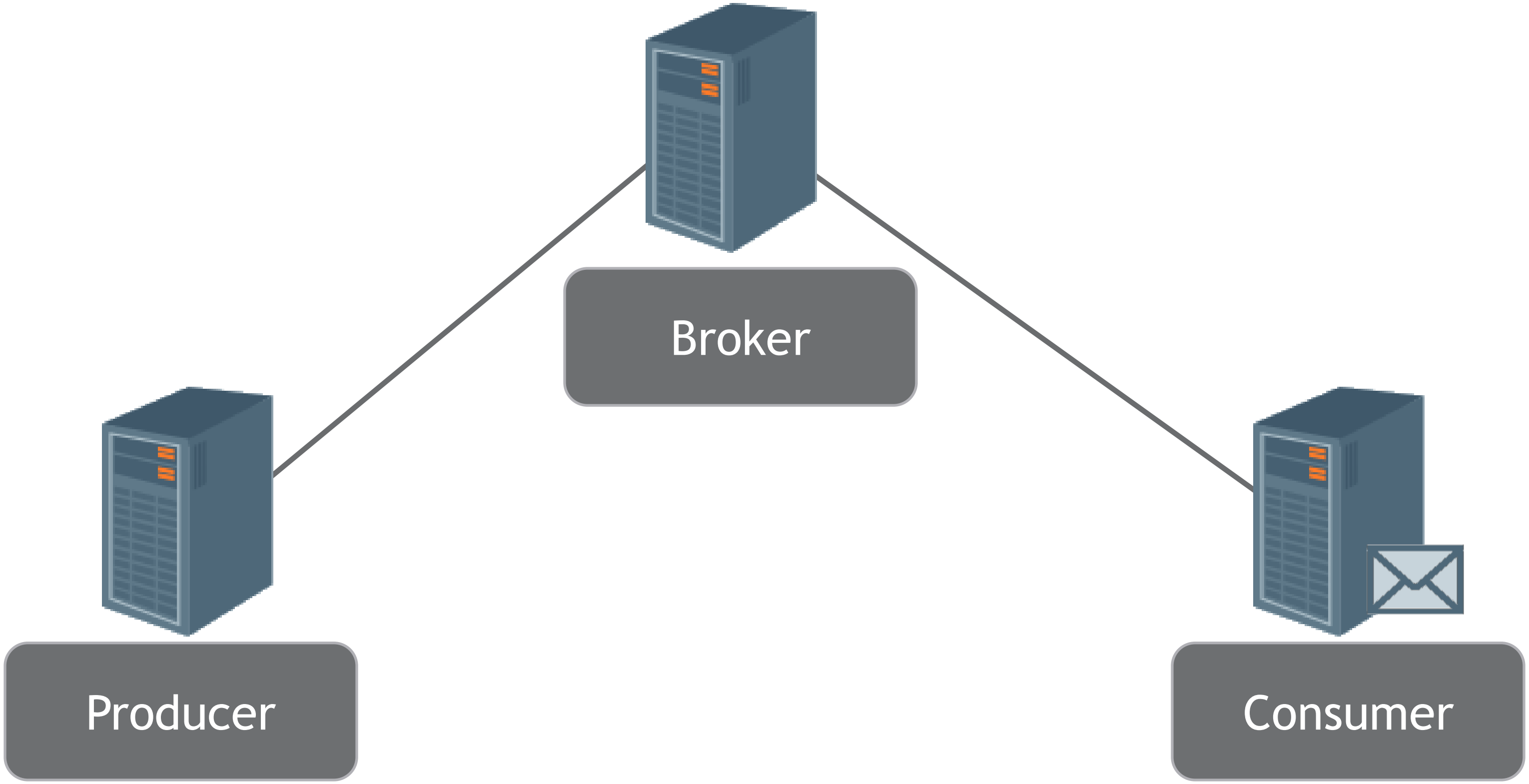
MQ Concepts



MQ Concepts



MQ Concepts



MQ Protocols

MQ Protocols

- MQ Transport Protocols
  - TCP, UDP, HTTP

MQ Protocols

- MQ Transport Protocols
  - TCP, UDP, HTTP
- MQ Application Protocols

MQ Protocols

- MQ Transport Protocols
  - TCP, UDP, HTTP
- MQ Application Protocols
  - Binary Protocols:
    - AMQP (Advanced Message Queuing Protocol)
    - MQTT (MQ Telemetry Transport)
    - OpenWire

MQ Protocols

- MQ Transport Protocols
  - TCP, UDP, HTTP
- MQ Application Protocols
  - Binary Protocols:
    - AMQP (Advanced Message Queuing Protocol)
    - MQTT (MQ Telemetry Transport)
    - OpenWire
  - ASCII Protocols:
    - STOMP (Streaming Text Oriented Messaging Protocol)
    - XMPP

MQ Security

- Transport over SSL/TLS
- Authentication and Authorisation Mechanisms:
  - Certificates, Kerberos, LDAP, etc.
- Persistent Storage
  - SQL Databases
  - File Based Databases
- Redundant Brokers
  - Clustering
  - Broker Networks

Misconfigurations

- Default Administrative Credentials
- Management Interfaces Exposed
  - Java Management Extension (JMX)
  - Java Remote Method Invocation (RMI)
  - Java Debug Wire Protocol (JDWP)
- Default Queues
  - Anonymous Access
    - Publish
    - Subscribe

Demo

- ActiveMQ 5.6.0
  - Debian 7.5.0
  - Ubuntu 14.04.1
- Default Configuration
- Java Management Extension (JMX)
  - Custom script to identify RMI service endpoint via JMX.
  - RMI Registry endpoint is only locally exposed.*
  - Port forwarding to access the RMI service.
  - Deploying and executing a JAR payload.

Case Studies

- Sending Serialised Objects
- Sending System Commands
- Rendering Untrusted Messages in Administrative or Monitoring Consoles
    - Cross-Site Scripting
- Inserting Unsanitised Messages in Databases
    - SQL Injection

Attack Scenarios

- Attacker’s Perspective
  - Anonymous
  - Client
  - Broker

- Attacks
  - Man-in-the-Middle
  - Authentication Bypasses
  - Implementation Specific
  - DoS

Attack Scenarios

- Attacker’s Perspective
  - Anonymous
  - Client
  - Broker

- Attacks
  - Man-in-the-Middle
  - Authentication Bypasses
  - Implementation Specific
  - DoS

Anonymous vs. Client

Attack Scenarios

- Attacker’s Perspective
  - Anonymous
  - Client
  - Broker

- Attacks
  - Man-in-the-Middle
  - Authentication Bypasses
  - Implementation Specific
  - DoS

Anonymous vs. Client

Attack Scenarios

- Attacker’s Perspective
  - Anonymous
  - Client
  - Broker

Anonymous vs. Client

Client vs. Client

- Attacks
  - Man-in-the-Middle
  - Authentication Bypasses
  - Implementation Specific
  - DoS

Attack Scenarios

- Attacker’s Perspective
  - Anonymous
  - Client
  - Broker

Anonymous vs. Client

Client vs. Client

Client vs. Broker

- Attacks
  - Man-in-the-Middle
  - Authentication Bypasses
  - Implementation Specific
  - DoS

Attack Scenarios

- Attacker’s Perspective
  - Anonymous
  - Client
  - Broker
- Attacks
  - Man-in-the-Middle
  - Authentication Bypasses
  - Implementation Specific
  - DoS

Anonymous vs. Client

Client vs. Client

Client vs. Broker

Attack Scenarios

- Attacker’s Perspective
  - Anonymous
  - Client
  - Broker

- Attacks
  - Man-in-the-Middle
  - Authentication Bypasses
  - Implementation Specific
  - DoS

Anonymous vs. Client

Client vs. Client

Client vs. Broker

Broker vs. Client

Attack Scenarios

- Attacker's Perspective
  - Anonymous
  - Client
  - Broker

- Attacks
  - Man-in-the-Middle
  - Authentication Bypasses
  - Implementation Specific
  - DoS

Anonymous vs. Client

Client vs. Client

Client vs. Broker

Broker vs. Client

Broker vs. Broker

Bug Hunting

Bug Hunting

- Source Code Audit
  - Pattern Based

Bug Hunting

- Source Code Audit
  - Pattern Based
- Fuzzing
  - Stateless
    - Radamsa
  - Stateful
    - MITM Fuzzing
  - Patching

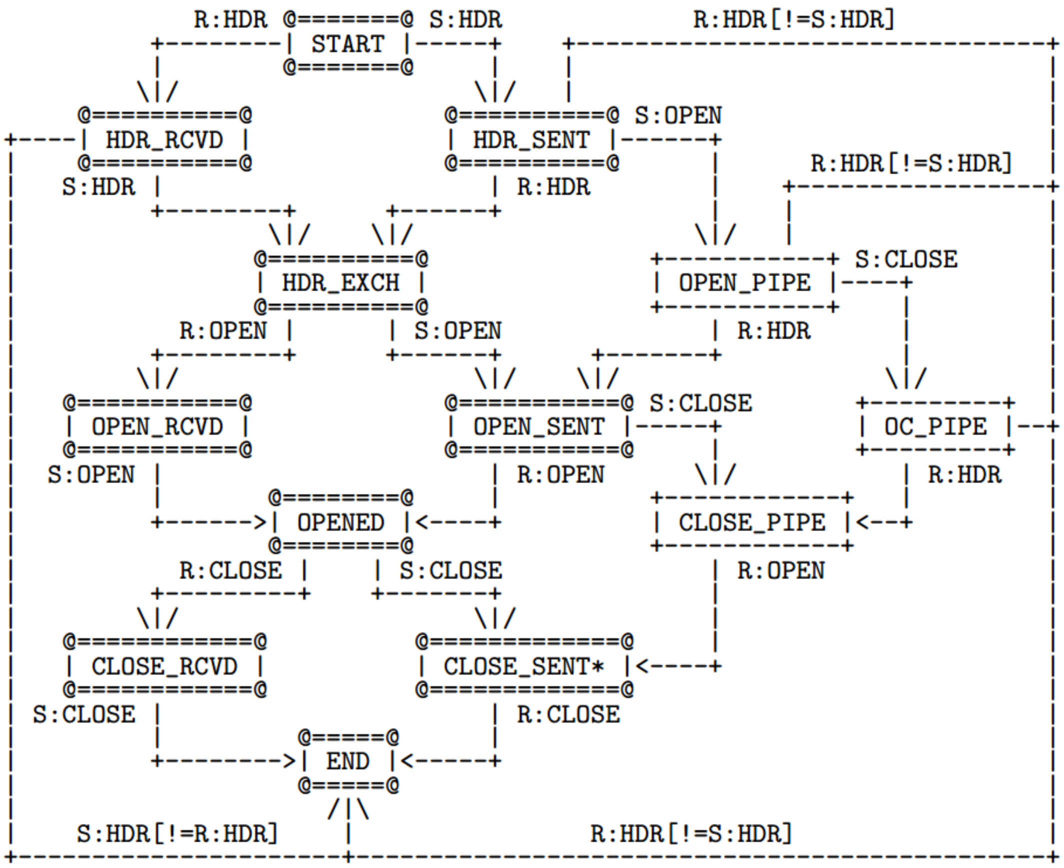
- Traffic Generation
  - Unit Tests
  - Performance Harness Tools
  - Code Samples

Bug Hunting

- Source Code Audit
  - Pattern Based
- Fuzzing
  - Stateless
    - Radamsa
  - Stateful
    - MITM Fuzzing
  - Patching
- Outdated Libraries
  - e.g. Vulnerable XStream in ActiveMQ < 5.10.0

- Traffic Generation
  - Unit Tests
  - Performance Harness Tools
  - Code Samples

AMQP State Machine

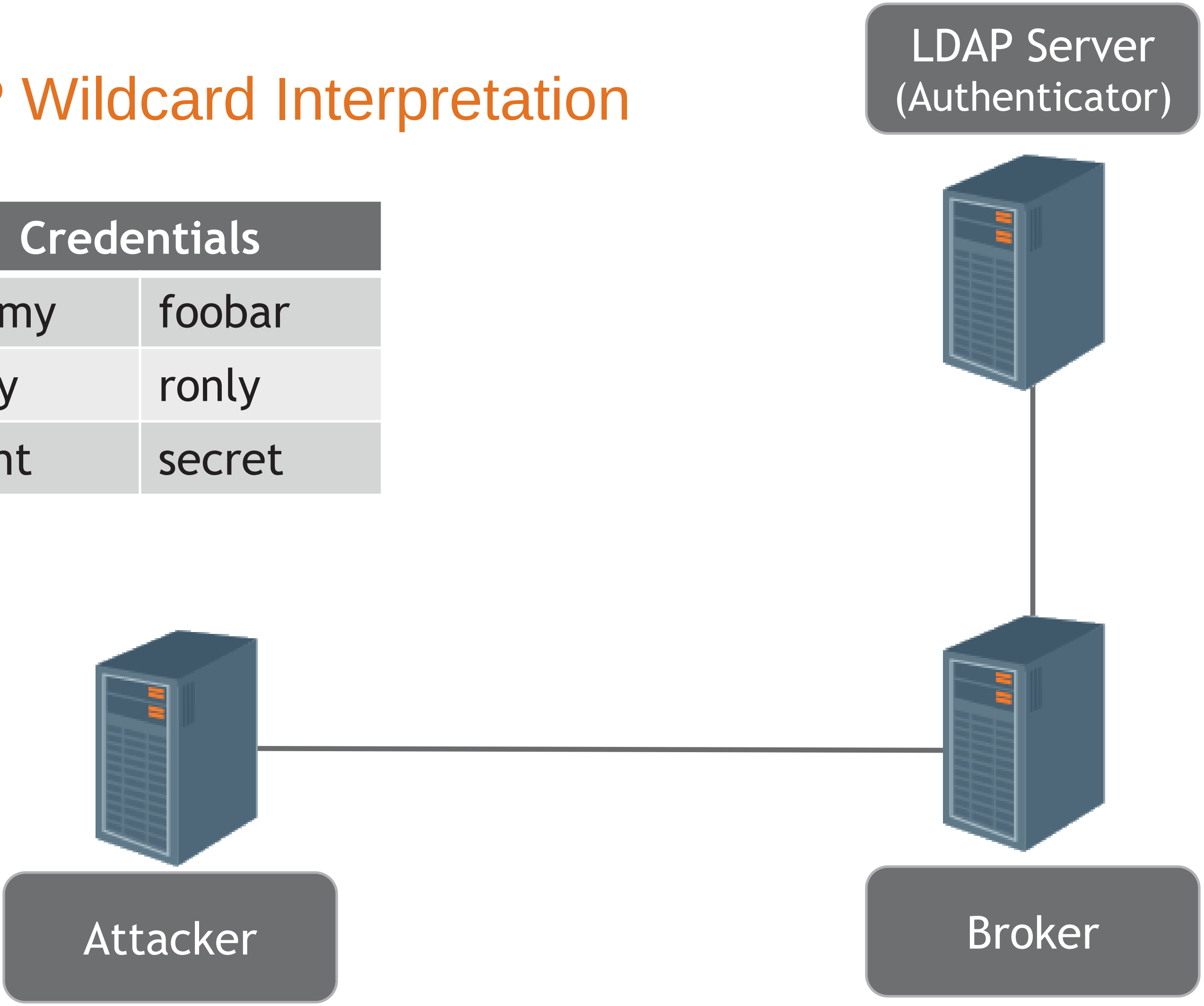


```
             R:HDR @=======@ S:HDR             R:HDR[!=S:HDR]
          +--------| START |-----+    +--------------------------------+
          |        @=======@     |    |                                |
         \|/                    \|/   |                                |
     @==========@             @==========@ S:OPEN                      |
+----| HDR_RCVD |             | HDR_SENT |------+                      |
|    @==========@             @==========@      |      R:HDR[!=S:HDR]  |
|   S:HDR |                      | R:HDR        |    +-----------------+
|         +--------+      +------+              |    |                 |
|                 \|/    \|/                   \|/   |                 |
|                @==========@               +-----------+ S:CLOSE      |
|                | HDR_EXCH |               | OPEN_PIPE |----+         |
|                @==========@               +-----------+    |         |
|           R:OPEN |      | S:OPEN              | R:HDR      |         |
|         +--------+      +------+      +-------+            |         |
|        \|/                    \|/    \|/                  \|/        |
|   @===========@             @===========@ S:CLOSE       +---------+  |
|   | OPEN_RCVD |             | OPEN_SENT |-----+         | OC_PIPE |--+
|   @===========@             @===========@     |         +---------+  |
|  S:OPEN |                      | R:OPEN      \|/           | R:HDR   |
|         |       @========@     |          +------------+   |         |
|         +------->| OPENED |<----+          | CLOSE_PIPE |<--+         |
|                 @========@                +------------+             |
|           R:CLOSE |    | S:CLOSE              | R:OPEN               |
|         +---------+    +-------+              |                      |
|        \|/                    \|/             |                      |
|   @============@          @=============@     |                      |
|   | CLOSE_RCVD |          | CLOSE_SENT* |<----+                      |
|   @============@          @=============@                            |
| S:CLOSE |                      | R:CLOSE                             |
|         |         @=====@      |                                     |
|         +--------->| END |<-----+                                     |
|                   @=====@                                            |
|                     /|\                                              |
|    S:HDR[!=R:HDR]    |                R:HDR[!=S:HDR]                 |
+----------------------+-----------------------------------------------+
```

```
R:<CTRL> = Received <CTRL>
S:<CTRL> = Sent <CTRL>
* Also could be DISCARDING if an error condition
  triggered the CLOSE
```

LDAP Wildcard Interpretation

Credentials	
tommy	foobar
ronly	ronly
client	secret



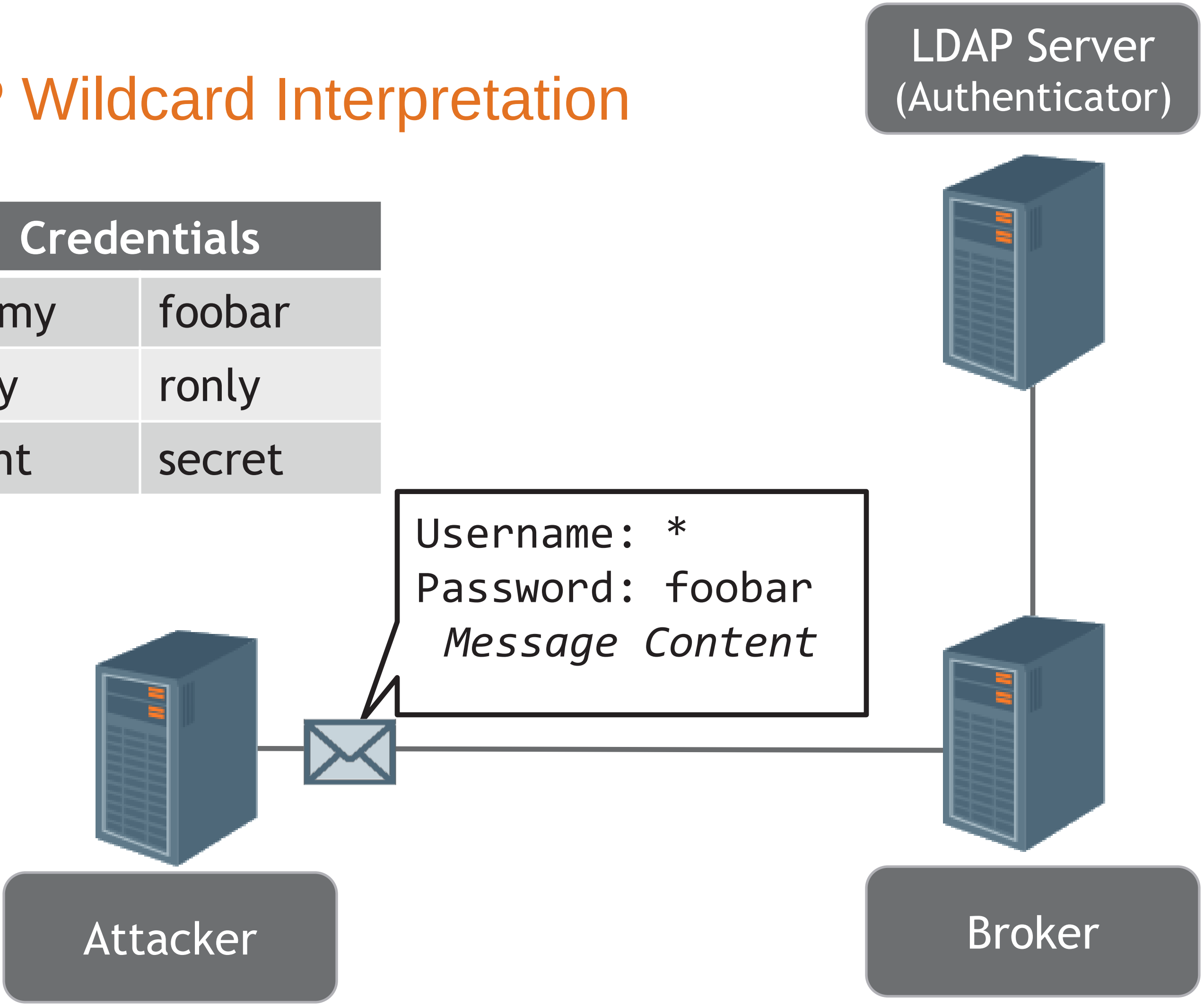
LDAP Server (Authenticator)

Attacker

Broker

LDAP Wildcard Interpretation

Credentials	
tommy	foobar
ronly	ronly
client	secret

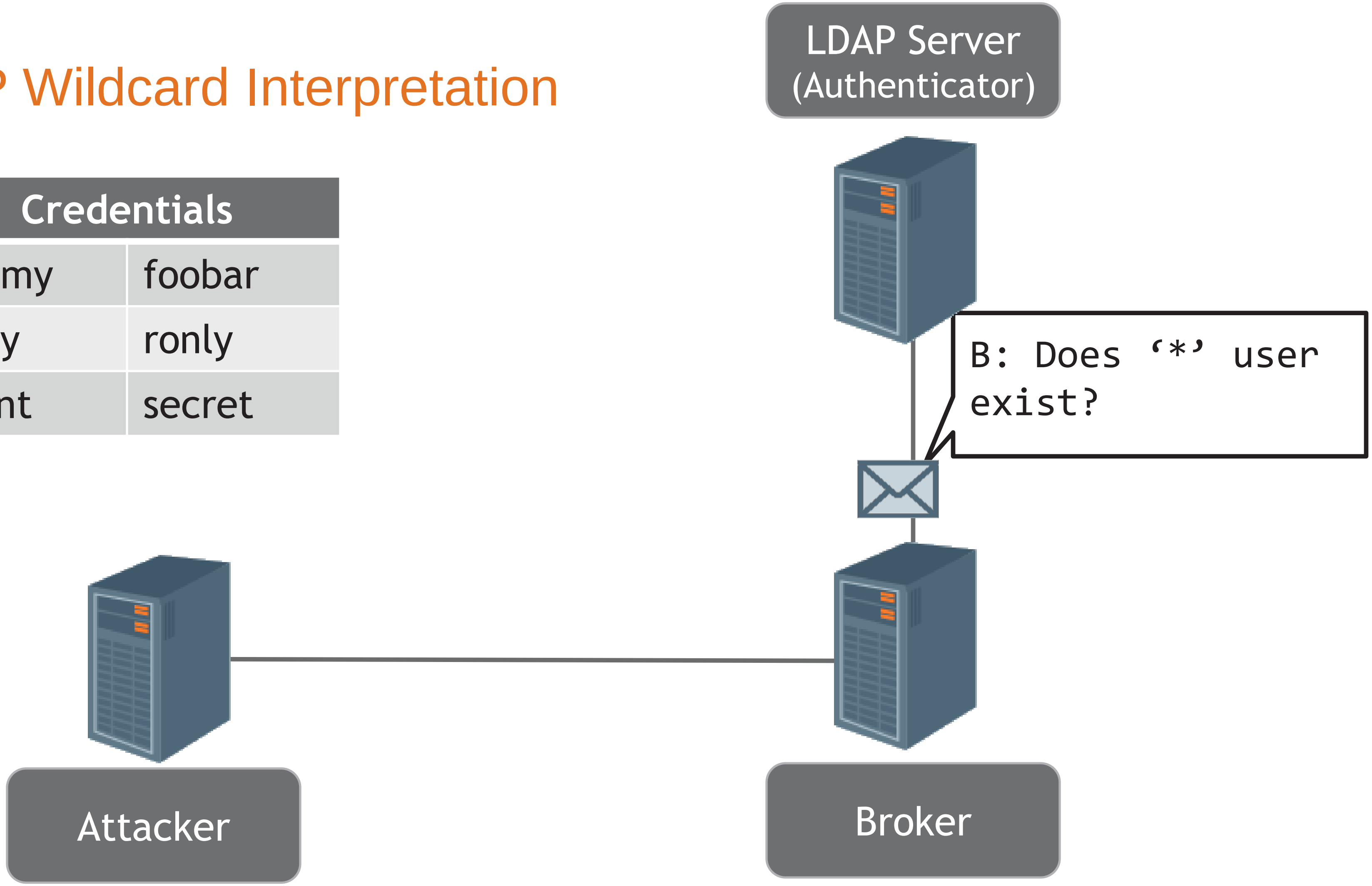


```
Username: *
Password: foobar
```

Message Content

LDAP Wildcard Interpretation

Credentials	
tommy	foobar
ronly	ronly
client	secret



LDAP Server (Authenticator)

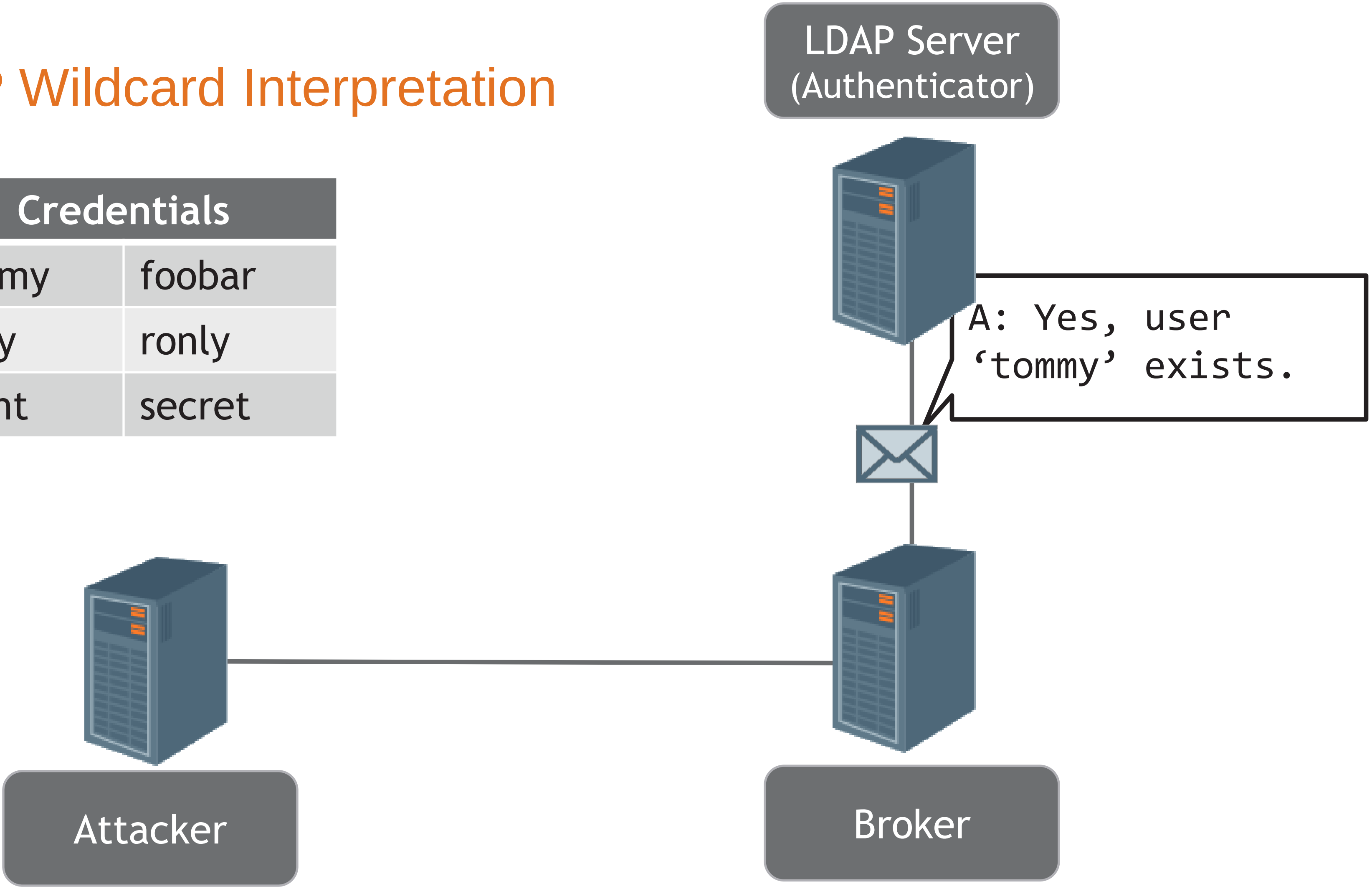
B: Does ‘*’ user exist?

Attacker

Broker

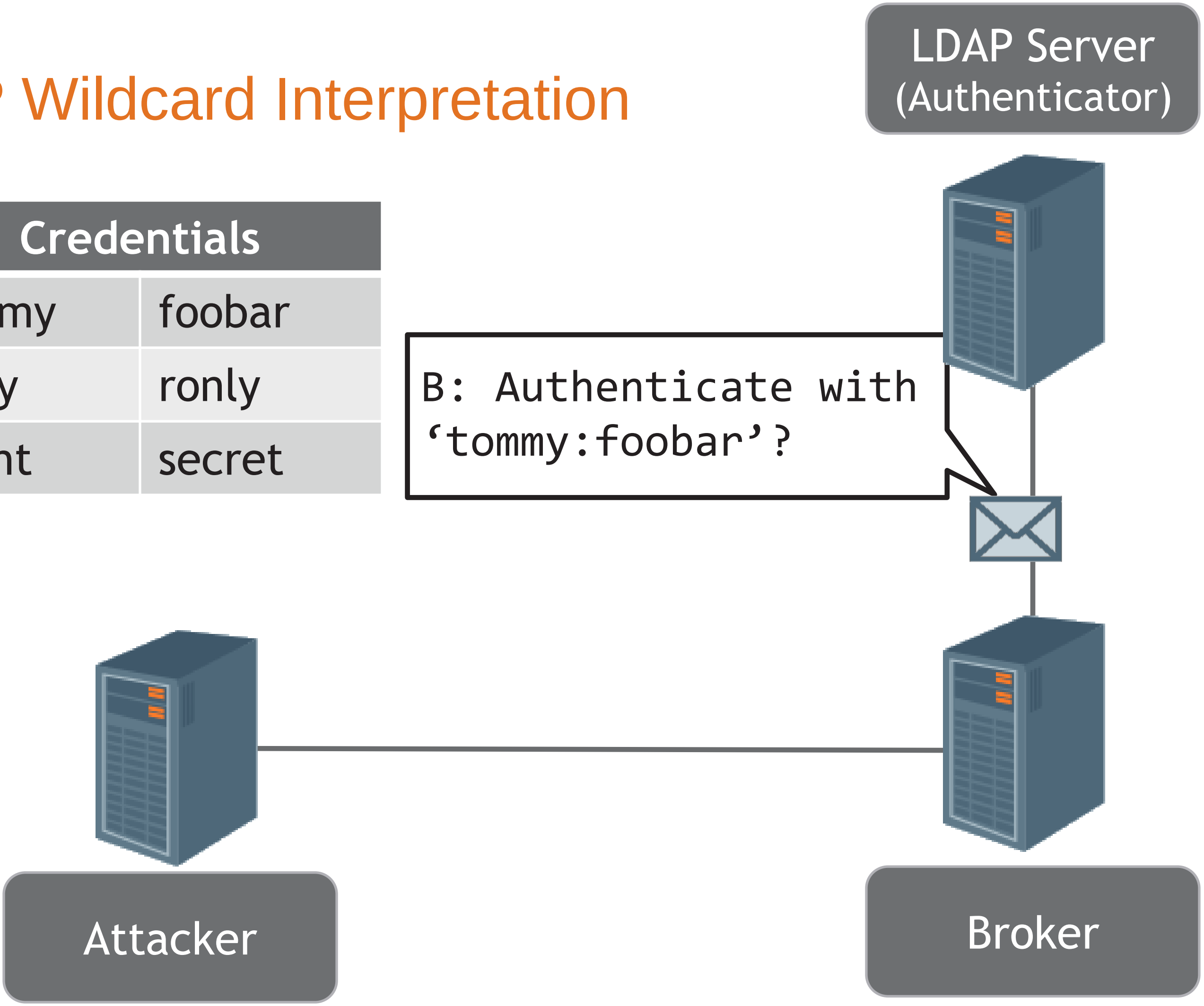
LDAP Wildcard Interpretation

Credentials	
tommy	foobar
ronly	ronly
client	secret



LDAP Wildcard Interpretation

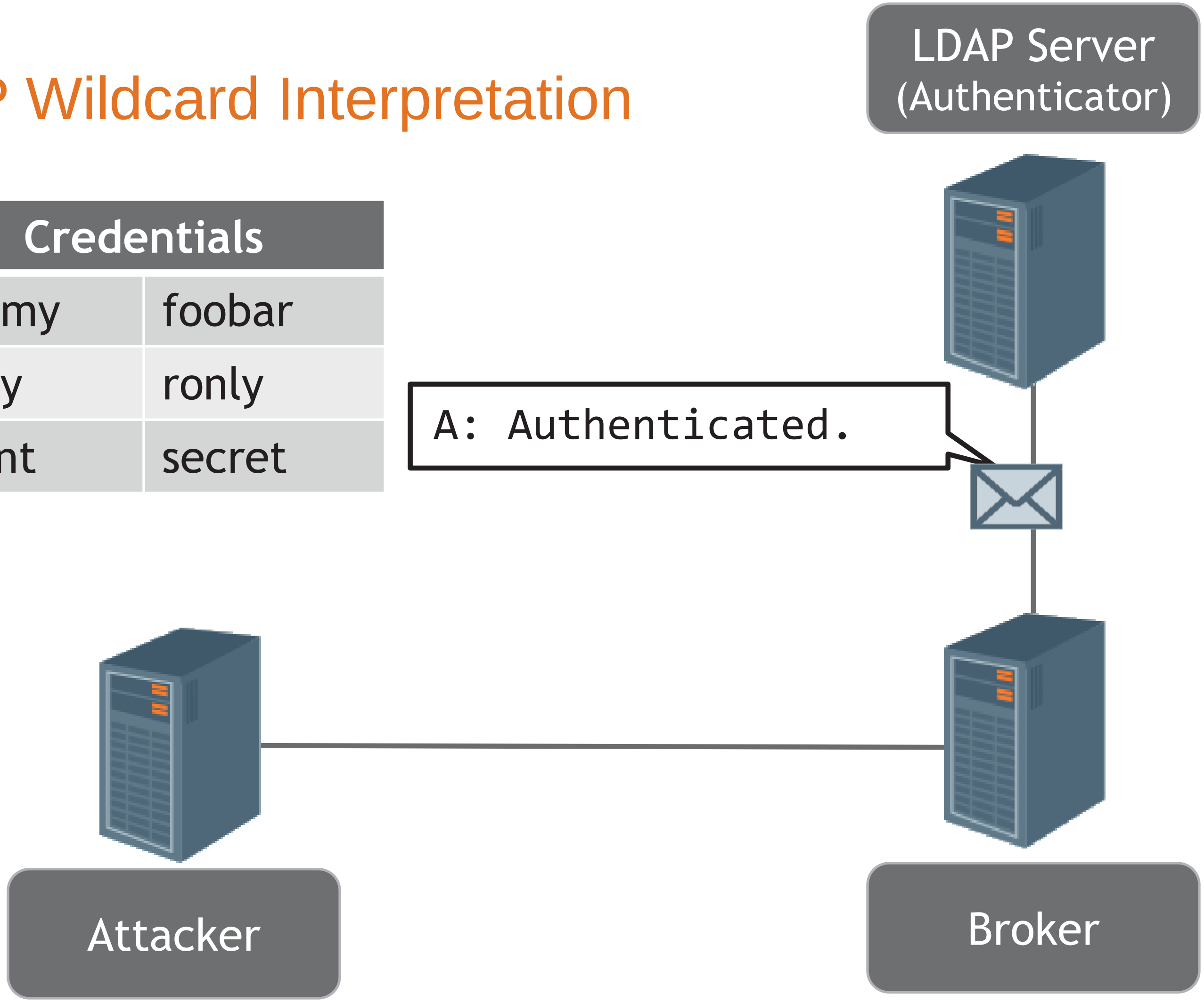
Credentials	
tommy	foobar
ronly	ronly
client	secret



B: Authenticate with ‘tommy:foobar’?

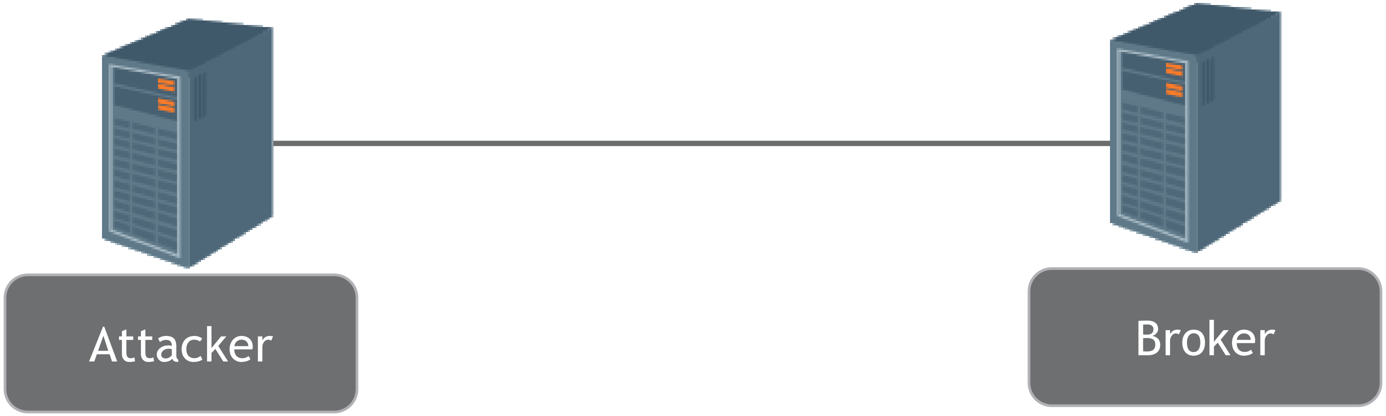
LDAP Wildcard Interpretation

Credentials	
tommy	foobar
ronly	ronly
client	secret

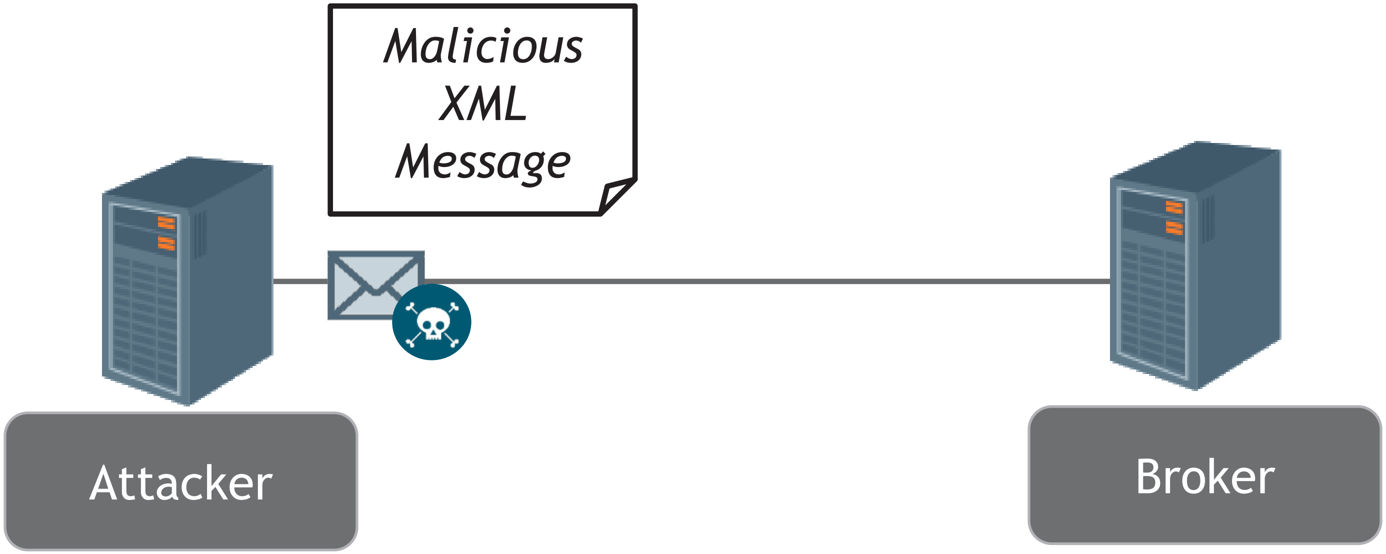


```
A: Authenticated.
```

XML External Entities Processing

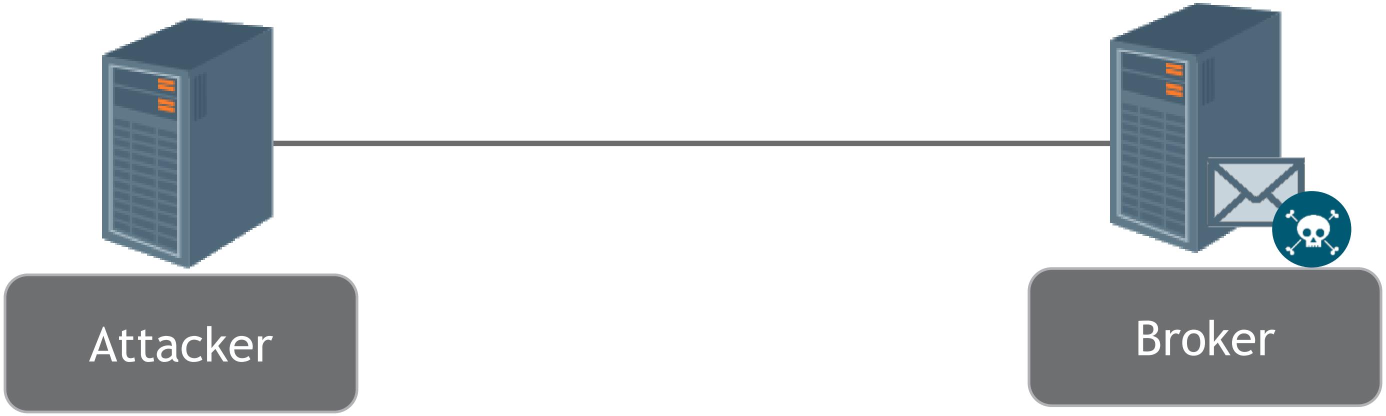


XML External Entities Processing



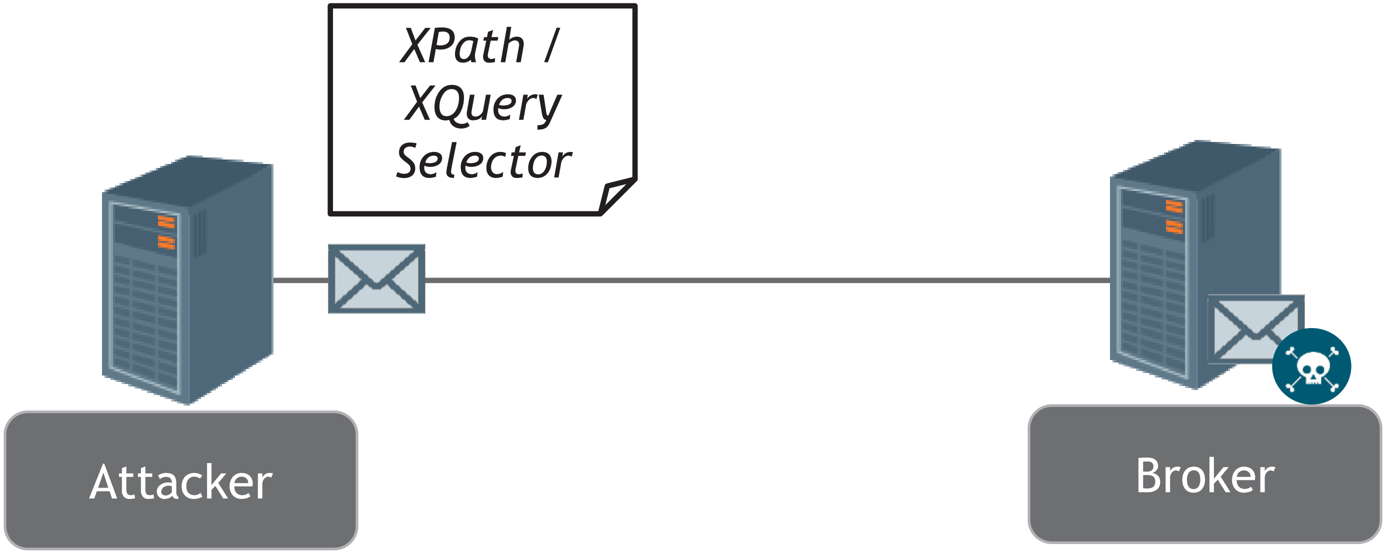
1. Adversary enqueues an XML message which contains XML external entities.

XML External Entities Processing



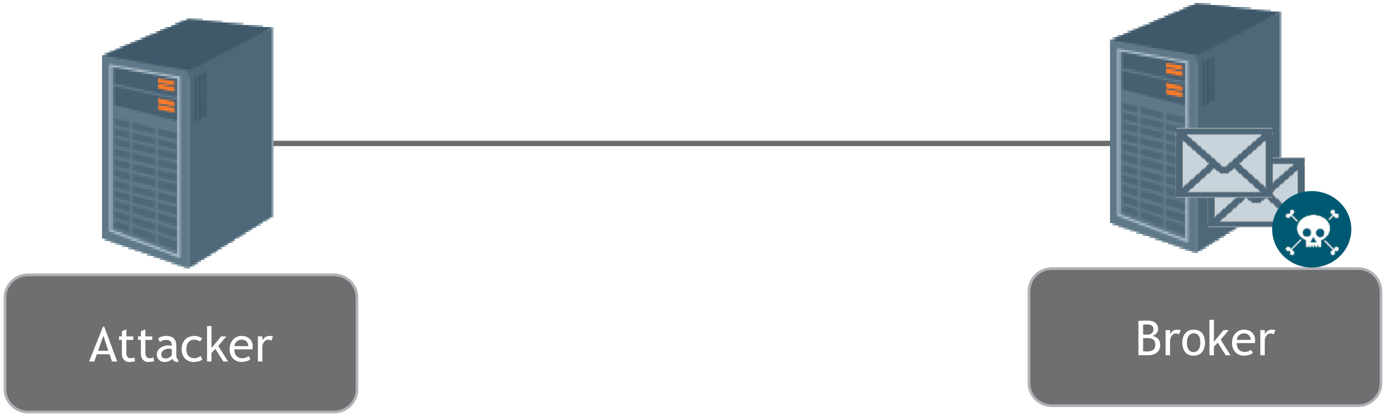
1. Adversary enqueues an XML message which contains XML external entities.

XML External Entities Processing



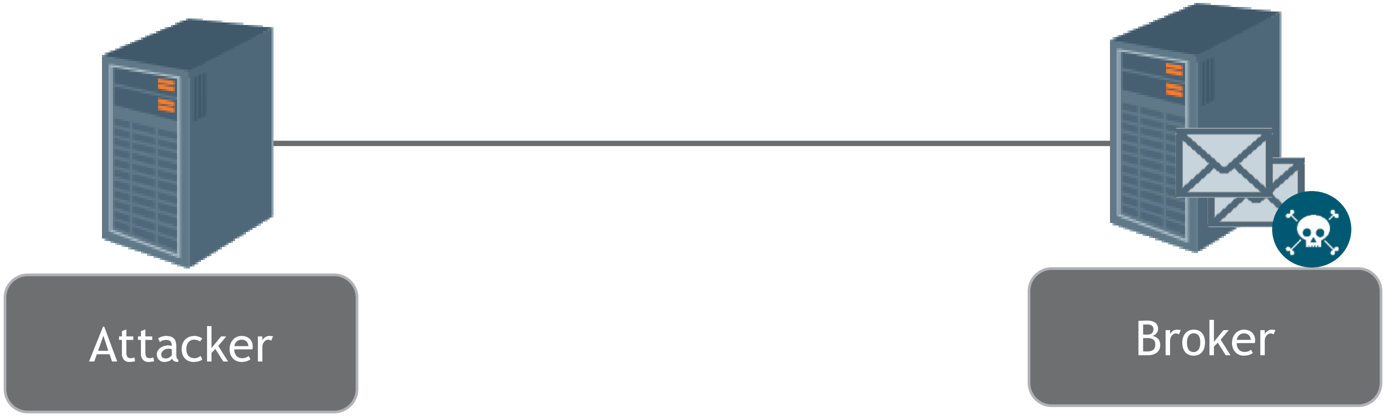
1. Adversary enqueues an XML message which contains XML external entities.
2. Then requests dequeuing an XML message which matches a criteria expressed with XPath/XQuery based selector.

XML External Entities Processing



1. Adversary enqueues an XML message which contains XML external entities.
2. Then requests dequeuing an XML message which matches a criteria expressed with XPath/XQuery based selector.

XML External Entities Processing



1. Adversary enqueues an XML message which contains XML external entities.
2. Then requests dequeuing an XML message which matches a criteria expressed with XPath/XQuery based selector.
3. The broker will evaluate the XPath expression and attempt to match it against the messages in the queue. This will cause the broker to resolve any external entity references.

Demo (1)

- Anonymous vs. Client / Broker
  - Authentication Bypass*

Demo (2)

- Client vs. Broker
  - XML External Entity Processing

Common Vulnerabilities

- XML External Entities Processing
  - Brokers: **6**
    - Java, Python and C++
  - Clients: 2*
- LDAP Wildcard Interpretation Bug
  - Brokers: **3**
    - Java
- Unserialisation of Untrusted Data
  - Brokers: 2*
    - Java and Python

Hardening

MQ

Applications

Hardening

MQ

- Limit the number of transport and application protocols.
  - One application protocol over one (SSL) transport.

Applications

Hardening

MQ

- Limit the number of transport and application protocols.
  - One application protocol over one (SSL) transport.
- Remove default accounts.

Applications

Hardening

MQ

- Limit the number of transport and application protocols.
  - One application protocol over one (SSL) transport.
- Remove default accounts.
- Disable JMX/RMI/JDWP/etc.*

Applications

Hardening

MQ

- Limit the number of transport and application protocols.
  - One application protocol over one (SSL) transport.
- Remove default accounts.
- Disable JMX/RMI/JDWP/etc.*
- Separate administration VLAN.

Applications

Hardening

MQ

Applications

- Limit the number of transport and application protocols.
  - One application protocol over one (SSL) transport.
- Remove default accounts.
- Disable JMX/RMI/JDWP/etc.*
- Separate administration VLAN.
- Disable anonymous client access.

Hardening

MQ

- Limit the number of transport and application protocols.
  - One application protocol over one (SSL) transport.
- Remove default accounts.
- Disable JMX/RMI/JDWP/etc.*
- Separate administration VLAN.
- Disable anonymous client access.
- Whitelist explicit P&S client IP addresses.

Applications

Hardening

MQ

- Limit the number of transport and application protocols.
  - One application protocol over one (SSL) transport.
- Remove default accounts.
- Disable JMX/RMI/JDWP/etc.*
- Separate administration VLAN.
- Disable anonymous client access.
- Whitelist explicit P&S client IP addresses.

Applications

- Perform validation on received messages. Do not assume trusted sources.

Hardening

MQ

- Limit the number of transport and application protocols.
  - One application protocol over one (SSL) transport.
- Remove default accounts.
- Disable JMX/RMI/JDWP/etc.*
- Separate administration VLAN.
- Disable anonymous client access.
- Whitelist explicit P&S client IP addresses.

Applications

- Perform validation on received messages. Do not assume trusted sources.
- Enable integrity checking. Ideally, authenticated encryption.

Hardening

MQ

- Limit the number of transport and application protocols.
  - One application protocol over one (SSL) transport.
- Remove default accounts.
- Disable JMX/RMI/JDWP/etc.*
- Separate administration VLAN.
- Disable anonymous client access.
- Whitelist explicit P&S client IP addresses.

Applications

- Perform validation on received messages. Do not assume trusted sources.
- Enable integrity checking. Ideally, authenticated encryption.
- Whitelist objects if unserialising from messages.

Acknowledgments

- MWR Labs
- Red Hat and Apache’s Security Teams
- NoSuchCon Organisers

References

- XML Out-of-Band Data Retrieval (BlackHat Europe 2013)
  - Timur Yunusov (@a66at)
  - Alexey Osipov (@Gi_sUngiven)
- XML External Entities Out-of-Band Exploitation
  - Ivan Novikov (@d0znpp)
- Exploiting JMX RMI
  - Braden Thomas

Questions

- Feedback
  - @munmap
  - georgi.geshev @ mwrinfosecurity . com