

Zielstellung und Rahmenbedingungen

Testumgebung

Verwendete bzw. Erstellte Programme

Beschreibung der Anwendungsszenarien

S1 Direkte Nachrichten 1:1

Beschreibung Ein Producer erzeugt kontinuierlich Nachrichten, die von einem Consumer kontinuierlich korrekt entnommen werden.

Aktion `perfTest -h`

Anmerkungen

S2

Beschreibung

Aktion

Anmerkungen

S3

Beschreibung

Aktion

Anmerkungen

Beschreibung der Angriffe

A1 Ignorieren von Nachrichten

<i>Beschreibung</i>	Ein Producer erzeugt kontinuierlich Nachrichten, die von einem oder mehreren Consumer empfangen, aber nicht quittiert werden. Der RabbitMQ-Server ist somit gezwungen, die Nachrichten in der Queue zwischenzuspeichern.
<i>Testparameter</i>	
<i>Befehlszeile</i>	<code>Amqpstress -dm No -c 5 -i 1 -ms 1048576 -mp -u amqp://testc:testp@localhost:5672/%2f</code>
<i>Beobachtungen</i>	
<i>Anmerkungen</i>	Durch Verwendung einer höheren Anzahl an Consumer, kleineren Sendeintervallen, größeren Nachrichtengrößen sowie durch Verwendung von persistenten Queues kann die Auswirkung des Angriffs erhöht werden.

A2 Sofortiges Abweisen von Nachrichten

<i>Beschreibung</i>	Ein Producer erzeugt kontinuierlich Nachrichten, die von einem oder mehreren Consumer empfangen, aber sofort abgewiesen (basic.Reject) werden. Der RabbitMQ-Server ist somit gezwungen, die Nachrichten in der Queue zwischenzuspeichern und erneut an den Consumer zu senden.
<i>Testparameter</i>	
<i>Befehlszeile</i>	<code>Amqpstress -dm REJECT -c 5 -i 1 -ms 1048576 -mp -u amqp://testc:testp@localhost:5672/%2f</code>
<i>Beobachtungen</i>	
<i>Anmerkungen</i>	Durch Verwendung einer höheren Anzahl an Consumer, kleineren Sendeintervallen, größeren Nachrichtengrößen sowie durch Verwendung von persistenten Queues kann die Auswirkung des Angriffs erhöht werden.

A3**Gebündeltes Abweisen von Nachrichten**

Beschreibung Ein Producer erzeugt kontinuierlich Nachrichten, die von einem oder mehreren Consumer empfangen, zunächst ignoriert werden, um sie bei Erreichen eines Schwellwertes gebündelt abzuweisen (basic.NACK). Dadurch ist der RabbitMQ-Server gezwungen, alle Nachrichten zwischenspeichern und stoßweise alle Nachrichten bis zu einer gewissen Nachrichtenkennung erneut zuzustellen.

Testparameter

Befehlszeile

```
Amqpstress -dm NACK -c 5 -i 1 -ms 1048576 -mp -u  
amqp://testc:testp@localhost:5672/%2f
```

Beobachtungen

Anmerkungen

Auswirkungen der Angriffe

Zusammenfassung und Fazit