Threat Modeling Report

Created on 11/13/2019 6:52:38 PM

Threat Model Name:

Owner:

Reviewer:

Contributors:

Description:

Assumptions:

External Dependencies:

Threat Model Summary:

Not Started	0
Not Applicable	0
Needs Investigation	0
Mitigation Implemented	30
Total	30
Total Migrated	0

Diagram: Diagram 1

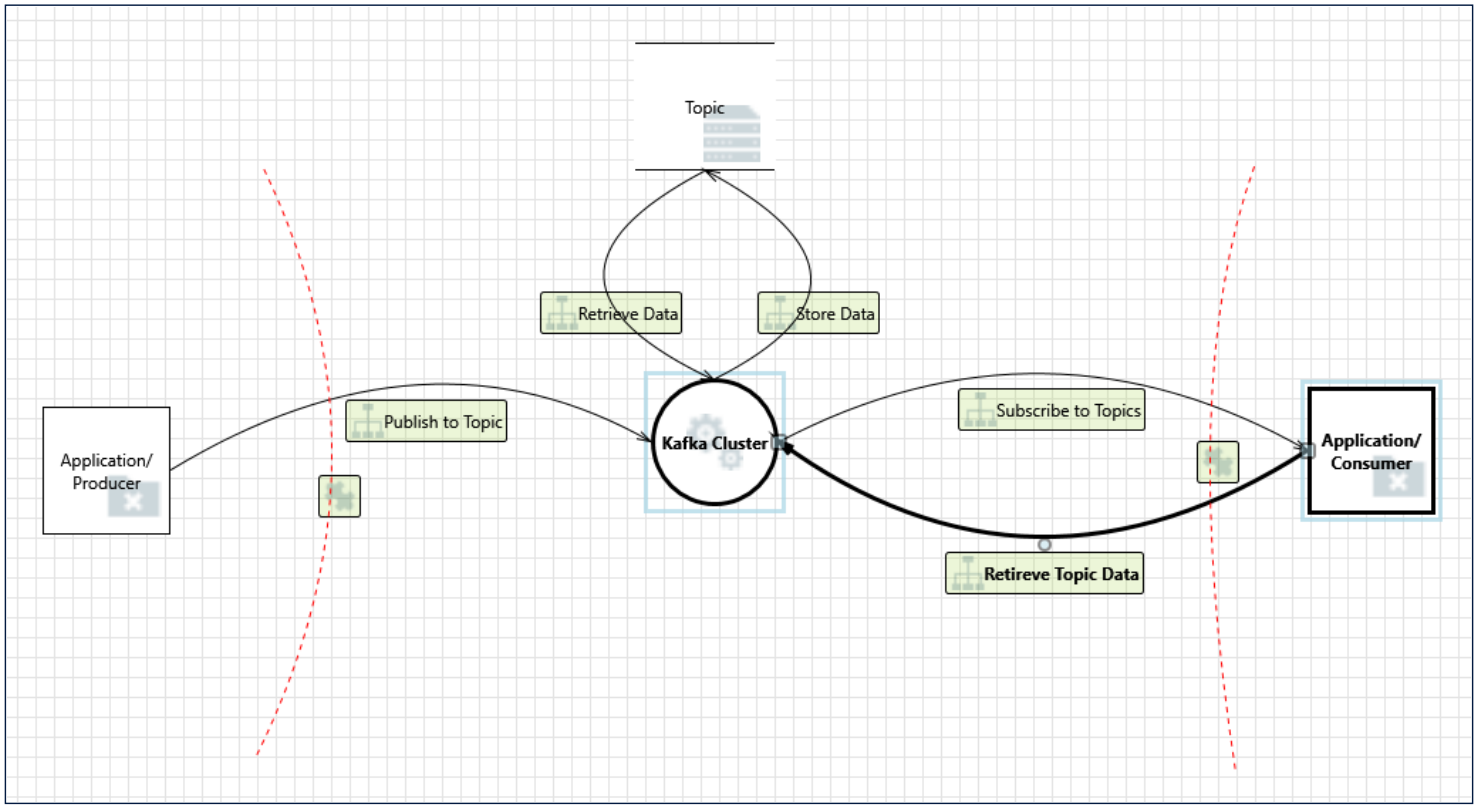


Diagram 1 Diagram Summary:

Not Started	0

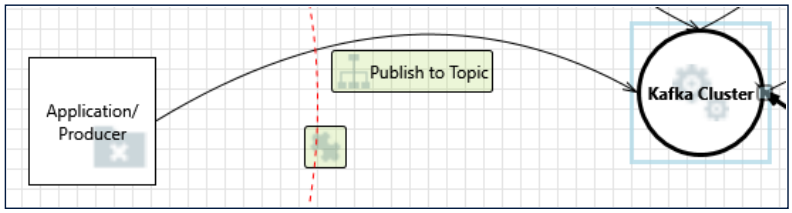
Not Applicable	0
Needs Investigation	0
Mitigation Implemented	30
Total	30
Total Migrated	0

Threat(s) Not Associated With an Interaction:

1. External Entity Administrator Potentially Denies Receiving Data [State: Mitigation Implemented] [Priority: High]

Category: Repudiation
Description: Administrator claims that it did not receive data from a process on the other side of the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data.
Justification: <no mitigation provided>

Interaction: Publish to Topic



2. Potential Data Repudiation by Broker [State: Mitigation Implemented] [Priority: High]

Category: Repudiation
Description: Kafka Cluster claims that it did not receive data from a source outside the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data.
Justification: Yammer Metrics is used to monitor activity in the server.

3. Potential Lack of Input Validation for Broker [State: Mitigation Implemented] [Priority: High]

Category: Tampering
Description: Data flowing across Publish to Topic may be tampered with by an attacker. This may lead to a denial of service attack against Kafka Cluster or an elevation of privilege attack against Kafka Cluster or an information disclosure by Kafka Cluster. Failure to verify that input is as expected is a root cause of a very large number of exploitable issues. Consider all paths and the way they handle data. Verify that all input is verified for correctness using an approved list input validation approach.
Justification: Kafka access control lists prevent unauthrorized producers from publishing/modifying data for a topic.

4. Spoofing the Application/Producer External Entity [State: Mitigation Implemented] [Priority: High]

Category: Spoofing
Description: Application/Producer may be spoofed by an attacker and this may lead to unauthorized access to Kafka Cluster. Consider using a standard authentication mechanism to identify the external entity.
Justification: Kafka requires client authentication and sets appropriate permissions per client.

5. Spoofing the Broker Process [State: Mitigation Implemented] [Priority: High]

Category: Spoofing
Description: Kafka Cluster may be spoofed by an attacker and this may lead to information disclosure by Application/Producer. Consider using a standard authentication mechanism to identify the destination process.
Justification: Kafka deploys SSL key/certificate pair for each broker of a cluster.

6. Cross Site Request Forgery [State: Mitigation Implemented] [Priority: High]

Category: Elevation Of Privilege
Description: Cross-site request forgery (CSRF or XSRF) is a type of attack in which an attacker forces a user's browser to make a forged request to a

vulnerable site by exploiting an existing trust relationship between the browser and the vulnerable web site. In a simple scenario, a user is logged in to web site A using a cookie as a credential. The other browses to web site B. Web site B returns a page with a hidden form that posts to web site A. Since the browser will carry the user's cookie to web site A, web site B now can take any action on web site A, for example, adding an admin to an account. The attack can be used to exploit any requests that the browser automatically authenticates, e.g. by session cookie, integrated authentication, IP whitelisting, ... The attack can be carried out in many ways such as by luring the victim to a site under control of the attacker, getting the user to click a link in a phishing email, or hacking a reputable web site that the victim will visit. The issue can only be resolved on the server side by requiring that all authenticated state-changing requests include an additional piece of secret payload (canary or CSRF token) which is known only to the legitimate web site and the browser and which is protected in transit through SSL/TLS. See the Forgery Protection property on the flow stencil for a list of mitigations.

Justification: Kafka is not a website, cross site request forgery N/A

7. Elevation by Changing the Execution Flow in Broker [State: Mitigation Implemented] [Priority: High]

Category: Elevation Of Privilege

Description: An attacker may pass data into Kafka Cluster in order to change the flow of program execution within Kafka Cluster to the attacker's choosing.

Justification: Kafka provides authentication and authorization using ACLs - need proper permissions - encryption in transit in place

8. Broker May be Subject to Elevation of Privilege Using Remote Code Execution [State: Mitigation Implemented] [Priority: High]

Category: Elevation Of Privilege

Description: Application/Producer may be able to remotely execute code for Kafka Cluster.

Justification: Kafka provides authentication and authorization using ACLs - need proper permissions - encryption in transit in place

9. Elevation Using Impersonation [State: Mitigation Implemented] [Priority: High]

Category: Elevation Of Privilege

Description: Kafka Cluster may be able to impersonate the context of Application/Producer in order to gain additional privilege.

Justification: Kafka provides authentication and authorization using ACLs - need proper permissions - encryption in transit in place

10. Data Flow Send Data Stream Is Potentially Interrupted [State: Mitigation Implemented] [Priority: High]

Category: Denial Of Service

Description: An external agent interrupts data flowing across a trust boundary in either direction.

Justification: This is mitigated by utilizing replicas, as well as through configuration. The configuration enable.idempotence=false allows for retries due to broker failure, and the retries configuration sets how many retries are allowed. If problems with stream interruption are a persistent threat, then interceptor.classes allows records to be intercepted (and possibly mutated) before they are published to the broker, allowing redirection to another broker or system if needed. Replicas allow producers and consumers to switch to another broker due to failure or connection interruption.

11. Potential Process Crash or Stop for Broker [State: Mitigation Implemented] [Priority: High]

Category: Denial Of Service

Description: Kafka Cluster crashes, halts, stops or runs slowly; in all cases violating an availability metric.

Justification: Broker failure is mitigated with the use of replicas and topic leaders managed by Zookeeper. There are several configurations that allow for fine grain control of timeouts, including transaction.timeout.ms and request.timeout.ms.

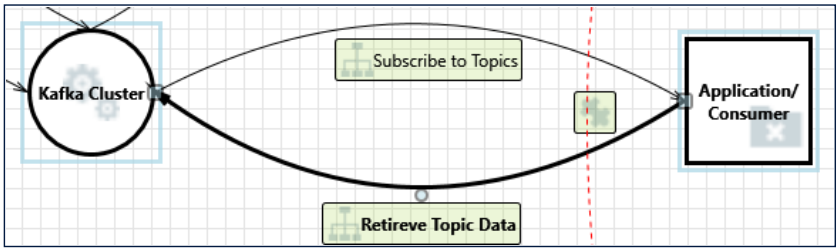
12. Data Flow Sniffing [State: Mitigation Implemented] [Priority: High]

Category: Information Disclosure

Description: Data flowing across Publish to Topic may be sniffed by an attacker. Depending on what type of data an attacker can read, it may be used to attack other parts of the system or simply be a disclosure of information leading to compliance violations. Consider encrypting the data flow.

Justification: Data transferred between brokers and clients can be encrypted using SSL.

Interaction: Retireve Topic Data



13. Cross Site Request Forgery [State: Mitigation Implemented] [Priority: High]

Category: Elevation Of Privilege

Description: Cross-site request forgery (CSRF or XSRF) is a type of attack in which an attacker forces a user's browser to make a forged request to a vulnerable site by exploiting an existing trust relationship between the browser and the vulnerable web site. In a simple scenario, a user is logged in to web site A using a cookie as a credential. The other browses to web site B. Web site B returns a page with a hidden form that posts to web site A. Since the browser will carry the user's cookie to web site A, web site B now can take any action on web site A, for example, adding an admin to an account. The attack can be used to exploit any requests that the browser automatically authenticates, e.g. by session cookie, integrated authentication, IP whitelisting, ... The attack can be carried out in many ways such as by luring the victim to a site under control of the attacker, getting the user to click a link in a phishing email, or hacking a reputable web site that the victim will visit. The issue can only be resolved on the server side by requiring that all authenticated state-changing requests include an additional piece of secret payload (canary or CSRF token) which is known only to the legitimate web site and the browser and which is protected in transit through SSL/TLS. See the Forgery Protection property on the flow stencil for a list of mitigations.

Justification: Not an external website - N/A

14. Elevation by Changing the Execution Flow in Broker [State: Mitigation Implemented] [Priority: High]

Category: Elevation Of Privilege

Description: An attacker may pass data into Kafka Cluster in order to change the flow of program execution within Kafka Cluster to the attacker's choosing.

Justification: Kafka provides authentication and authorization using ACLs - need proper permissions - encryption in transit in place

15. Broker May be Subject to Elevation of Privilege Using Remote Code Execution [State: Mitigation Implemented] [Priority: High]

Category: Elevation Of Privilege

Description: Application/Consumer may be able to remotely execute code for Kafka Cluster.

Justification: Kafka provides authentication and authorization using ACLs - need proper permissions - encryption in transit in place

16. Elevation Using Impersonation [State: Mitigation Implemented] [Priority: High]

Category: Elevation Of Privilege

Description: Kafka Cluster may be able to impersonate the context of Application/Consumer in order to gain additional privilege.

Justification: Kafka provides authentication and authorization using ACLs - need proper permissions - encryption in transit in place

17. Data Flow Request Topic Data Is Potentially Interrupted [State: Mitigation Implemented] [Priority: High]

Category: Denial Of Service

Description: An external agent interrupts data flowing across a trust boundary in either direction.

Justification: This interruption is mitigated by the use of replicas and various timeout configurations. If a broker is determined to be unreachable, then a replica will take its place as the leader, and continue normal operations.

18. Potential Process Crash or Stop for Broker [State: Mitigation Implemented] [Priority: High]

Category: Denial Of Service

Description: Kafka Cluster crashes, halts, stops or runs slowly; in all cases violating an availability metric.

Justification: Broker failure is mitigated with the use of replicas and topic leaders managed by Zookeeper. There are several configurations that allow for fine grain control of timeouts, including transaction.timeout.ms and request.timeout.ms.

19. Data Flow Sniffing [State: Mitigation Implemented] [Priority: High]

Category: Information Disclosure

Description: Data flowing across Retireve Topic Data may be sniffed by an attacker. Depending on what type of data an attacker can read, it may be used to attack other parts of the system or simply be a disclosure of information leading to compliance violations. Consider encrypting the data flow.

Justification: Data transferred between brokers and clients can be encrypted using SSL.

20. Potential Data Repudiation by Broker [State: Mitigation Implemented] [Priority: High]

Category: Repudiation

Description: Kafka Cluster claims that it did not receive data from a source outside the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data.

Justification: Yammer Metrics is used to monitor activity in the server.

21. Potential Lack of Input Validation for Broker [State: Mitigation Implemented] [Priority: High]

Category: Tampering

Description: Data flowing across Retireve Topic Data may be tampered with by an attacker. This may lead to a denial of service attack against Kafka Cluster or an elevation of privilege attack against Kafka Cluster or an information disclosure by Kafka Cluster. Failure to verify that input is as expected is a root cause of a very large number of exploitable issues. Consider all paths and the way they handle data. Verify that all input is verified for correctness using an approved list input validation approach.

Justification: Kafka access control lists prevent unauthrorized brokers from modifying data for a topic.

22. Spoofing the Application/Consumer External Entity [State: Mitigation Implemented] [Priority: High]

Category: Spoofing

Description: Application/Consumer may be spoofed by an attacker and this may lead to unauthorized access to Kafka Cluster. Consider using a standard authentication mechanism to identify the external entity.

Justification: Kafka requires client authentication and sets appropriate permissions per client.

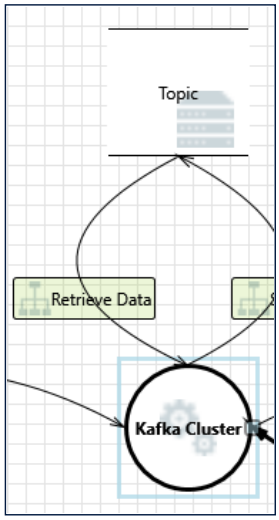
23. Spoofing the Broker Process [State: Mitigation Implemented] [Priority: High]

Category: Spoofing

Description: Kafka Cluster may be spoofed by an attacker and this may lead to information disclosure by Application/Consumer. Consider using a standard authentication mechanism to identify the destination process.

Justification: Kafka deploys SSL key/certificate pair for each broker of a cluster.

Interaction: Retrieve Data



24. Weak Access Control for a Resource [State: Mitigation Implemented] [Priority: High]

Category: Information Disclosure

Description: Improper data protection of Topic can allow an attacker to read information not intended for disclosure. Review authorization settings.

Justification: Data transferred between brokers (leader and replicas) can be encrypted using SSL.

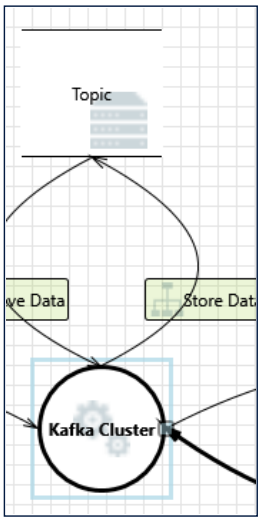
25. Spoofing of Source Data Store Topic [State: Mitigation Implemented] [Priority: High]

Category: Spoofing

Description: Topic may be spoofed by an attacker and this may lead to incorrect data delivered to Kafka Cluster. Consider using a standard authentication mechanism to identify the source data store.

Justification: Kafka requires authorization from topic resources for topic to be fetched.

Interaction: Store Data



26. Potential Excessive Resource Consumption for Kafka Cluster or Topic [State: Mitigation Implemented] [Priority: High]

Category: Denial Of Service

Description: Does Kafka Cluster or Topic take explicit steps to control resource consumption? Resource consumption attacks can be hard to deal with, and there are times that it makes sense to let the OS do the job. Be careful that your resource requests don't deadlock, and that they do timeout.

Justification: <no mitigation provided>

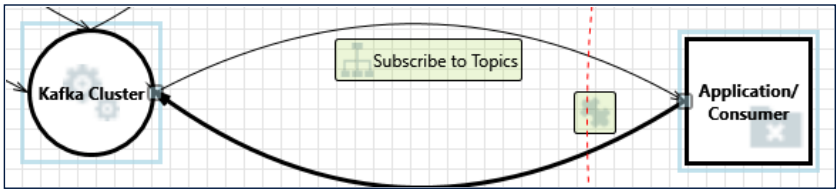
27. Spoofing of Destination Data Store Topic [State: Mitigation Implemented] [Priority: High]

Category: Spoofing

Description: Topic may be spoofed by an attacker and this may lead to data being written to the attacker's target instead of Topic. Consider using a standard authentication mechanism to identify the destination data store.

Justification: Kafka requires authorization from topic resources for topic to be fetched.

Interaction: Subscribe to Topics



28. Data Flow Send Topic Data Is Potentially Interrupted [State: Mitigation Implemented] [Priority: High]

Category: Denial Of Service

Description: An external agent interrupts data flowing across a trust boundary in either direction.

Justification: This interruption is mitigated by the use of replicas and various timeout configurations. If a broker is determined to be unreachable, then a

replica will take its place as the leader, and continue normal operations.

29. External Entity Application/Consumer Potentially Denies Receiving Data [State: Mitigation Implemented] [Priority: High]

Category: Repudiation

Description: Application/Consumer claims that it did not receive data from a process on the other side of the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data.

Justification: Kafka Metrics is used to monitor java client activity.

30. Spoofing of the Application/Consumer External Destination Entity [State: Mitigation Implemented] [Priority: High]

Category: Spoofing

Description: Application/Consumer may be spoofed by an attacker and this may lead to data being sent to the attacker's target instead of Application/Consumer. Consider using a standard authentication mechanism to identify the external entity.

Justification: Kafka requires client authentication and sets appropriate permissions per client.