Threat Modeling Report

Created on 11/14/2020 1:38:36 PM

Threat Model Name: Elasticsearch - Update Customer Account

Owner: Team 5 - CYBR8420-850:Software Assurance

Reviewer:

Contributors: Adrian Dorsey, Austin Vornhagen, Joseph Burr, Nicholas Palacio, Ryan Narducci

Description: A bank employee using a web client that communicates with an Elasticsearch cluster to update data in an index that holds customer account information

Assumptions:

External Dependencies:

Threat Model Summary:

Not Started	0
Not Applicable	14
Needs Investigation	20
Mitigation Implemented	26
Total	60
Total Migrated	0

Diagram: Diagram 2

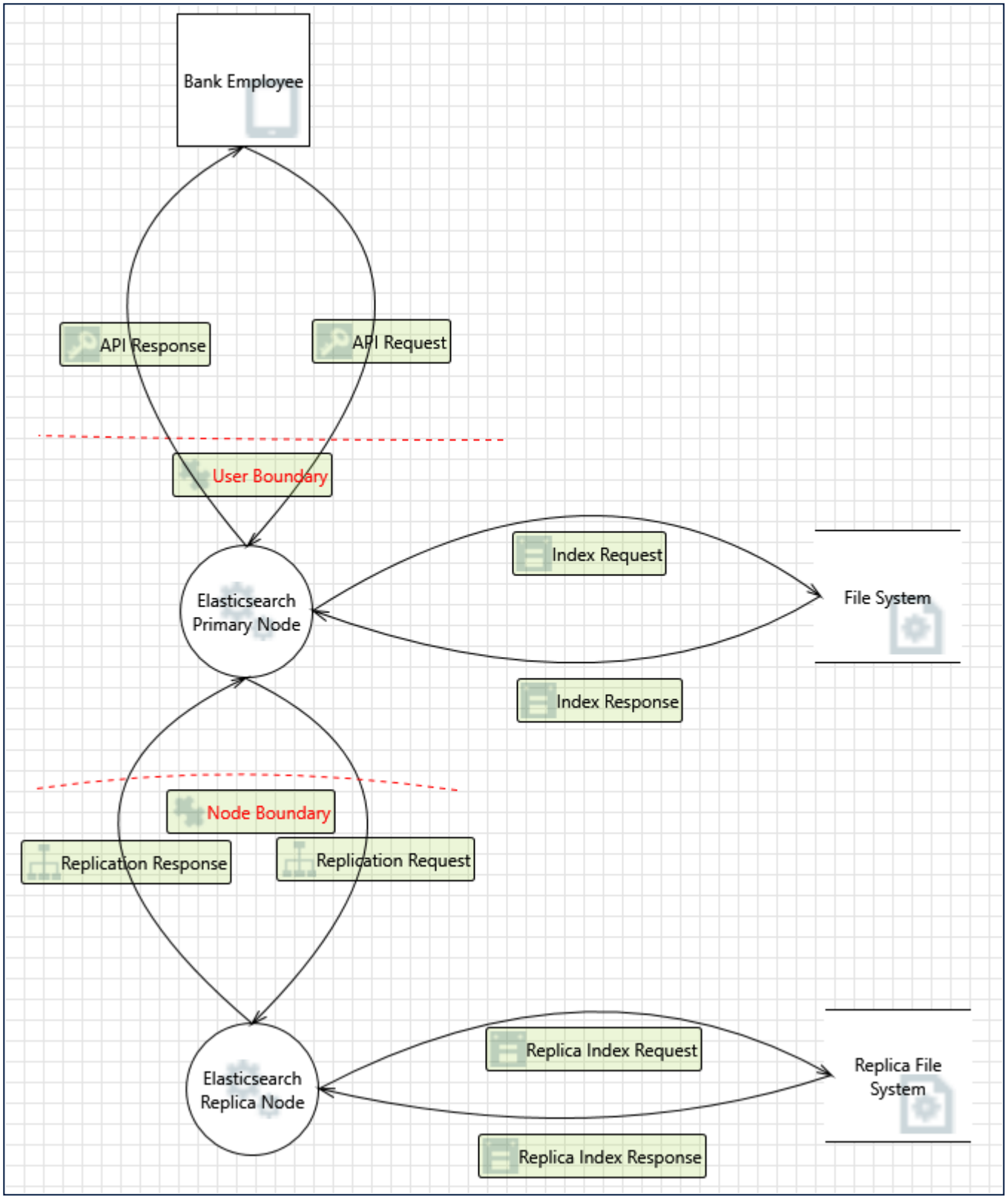
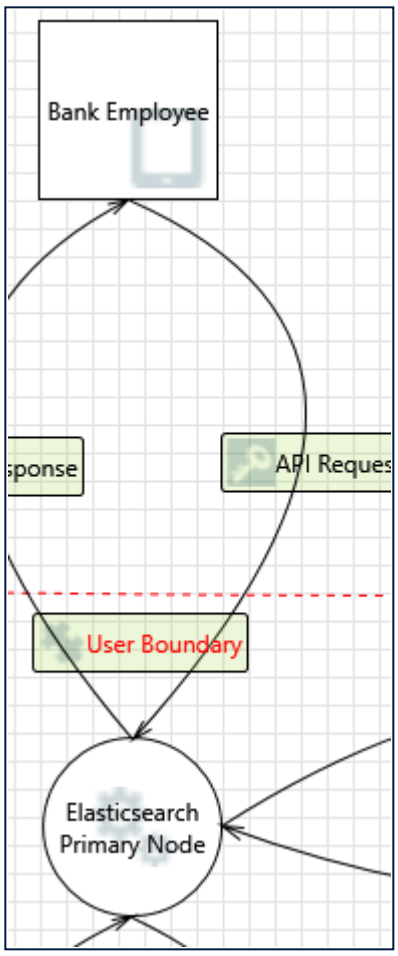


Diagram 2 Diagram Summary:

Not Started	0
Not Applicable	14
Needs Investigation	20
Mitigation Implemented	26

Total	60
Total Migrated	0

Interaction: API Request



1. Elevation by Changing the Execution Flow in Elasticsearch Primary Node [State: Needs Investigation] [Priority: High]

Category:	Elevation Of Privilege
Description:	An attacker may pass data into Elasticsearch Primary Node in order to change the flow of program execution within Elasticsearch Primary Node to the attacker's choosing.
Justification:	Further investigation is required to determine if program flow can be altered in harmful ways by new data.

2. Elasticsearch Primary Node May be Subject to Elevation of Privilege Using Remote Code Execution [State: Not Applicable] [Priority: High]

Category:	Elevation Of Privilege
Description:	Bank Employee may be able to remotely execute code for Elasticsearch Primary Node.
Justification:	Elatsicsearch's API does not allow for arbitrary remote code execution.

3. Elevation Using Impersonation [State: Not Applicable] [Priority: High]

Category: Elevation Of Privilege

Description: Elasticsearch Primary Node may be able to impersonate the context of Bank Employee in order to gain additional privilege.

Justification: Bank Employees represent users with access to a subset of data on the Elasticsearch nodes. If the Elasticsearch Primary Node is compromised and seeks to impersonate a user, the attacker already has access to the File System containing all of the data.

4. Data Flow API Request Is Potentially Interrupted [State: Needs Investigation] [Priority: High]

Category: Denial Of Service

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

Justification: Need to further investigate and determine our policy to follow if the Bank Employee cannot reach the Elasticsearch node. Perhaps physical backups of account information would need to be stored securely in the bank in the event that the portal is down.

5. Potential Process Crash or Stop for Elasticsearch Primary Node [State: Mitigation Implemented] [Priority: High]

Category: Denial Of Service

Description: Elasticsearch Primary Node crashes, halts, stops or runs slowly; in all cases violating an availability metric.

Justification: The Elasticsearch cluster is configured for resiliency. Configured with the recommending guidelines (See https://www.elastic.co/guide/en/elasticsearch/reference/current/high-availability-cluster-design.html), loss of the primary node can be handled.

6. Potential Data Repudiation by Elasticsearch Primary Node [State: Mitigation Implemented] [Priority: High]

Category: Repudiation

Description: Elasticsearch Primary Node 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: Elasticsearch supports security logs with the required event information (See https://www.elastic.co/guide/en/elasticsearch/reference/current/audit-event-types.html)

7. JavaScript Object Notation Processing [State: Needs Investigation] [Priority: High]

Category: Tampering

Description: If a dataflow contains JSON, JSON processing and hijacking threats may be exploited.

Justification: Further research of Elasticsearch's JSON processing is required.

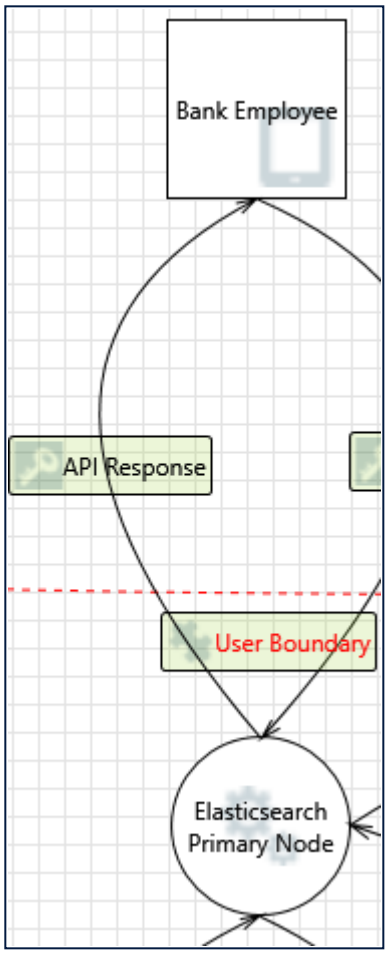
8. Spoofing the Bank Employee External Entity [State: Mitigation Implemented] [Priority: High]

Category: Spoofing

Description: Bank Employee may be spoofed by an attacker and this may lead to unauthorized access to Elasticsearch Primary Node. Consider using a standard authentication mechanism to identify the external entity.

Justification: Elasticsearch supports user authentcation, and it is being used (See https://www.elastic.co/guide/en/elasticsearch/reference/current/setting-up-authentication.html).

Interaction: API Response



9. Data Flow API Response Is Potentially Interrupted [State: Needs Investigation] [Priority: High]

Category: Denial Of Service

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

Justification: Need to further investigate and determine our policy to follow if the Elasticsearch node cannot respond back to the Bank Employee. The audit logs may log an event if a response cannot be sent back to a client. It would also be beneficial to figure out if Elasticsearch has a live alerting feature to notify administrators.

10. External Entity Bank Employee Potentially Denies Receiving Data [State: Mitigation Implemented] [Priority: High]

Category: Repudiation

Description: Bank Employee 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: Elasticsearch supports security logs with the required event information (See https://www.elastic.co/guide/en/elasticsearch/reference/current/audit-event-types.html)

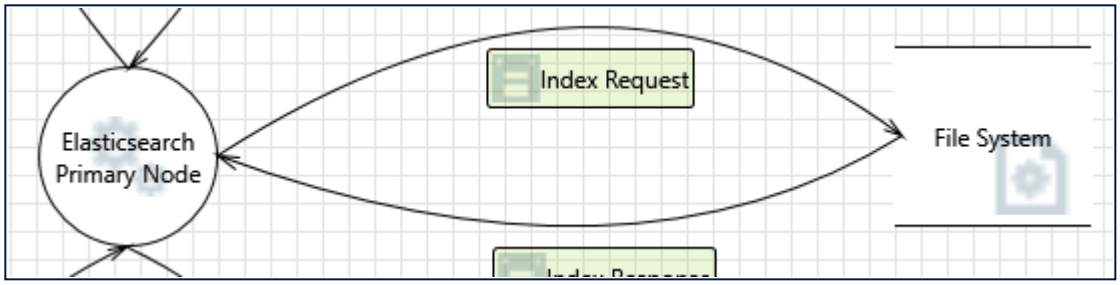
11. Spoofing of the Bank Employee External Destination Entity [State: Mitigation Implemented] [Priority: High]

Category: Spoofing

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

Justification: Elasticsearch supports user authentcation, and it is being used (See https://www.elastic.co/guide/en/elasticsearch/reference/current/setting-up-authentication.html). Additionally, the REST API can be configured to use HTTPS to mitigate man-in-the-middle attacks.

Interaction: Index Request



12. Potential Weak Protections for Audit Data [State: Mitigation Implemented] [Priority: High]

Category: Repudiation

Description: Consider what happens when the audit mechanism comes under attack, including attempts to destroy the logs, or attack log analysis programs. Ensure access to the log is through a reference monitor, which controls read and write separately. Document what filters, if any, readers can rely on, or writers should expect

Justification: Access to the File System via Elasticsearch is controlled by Elasticsearch's user authentication. Any other access the Elasticsearch log files for auditing purposes on each server will be mitigated using conventional role-based access control on the OS.

13. Potential Excessive Resource Consumption for Elasticsearch Primary Node or File System [State: Needs Investigation] [Priority: High]

Category: Denial Of Service

Description: Does Elasticsearch Primary Node or File System 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: Need to investigate if Elasticsearch has any bandwidth limiting for file system access.

14. Spoofing of Destination Data Store File System [State: Not Applicable] [Priority: High]

Category: Spoofing

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

Justification: The index files' paths are built into Elasticsearch. If an attacker has access to the File System, they have access to the machine that Elasticsearch is running on. This should be mitigated with OS role based access control outside of Elasticsearch.

15. Lower Trusted Subject Updates Logs [State: Not Applicable] [Priority: High]

Category: Repudiation

Description: If you have trust levels, is anyone other outside of the highest trust level allowed to log? Letting everyone write to your logs can lead to repudiation problems. Only allow trusted code to log.

Justification: Elasticsearch will generate logs for all users but users do not directly control the Elasticsearch logging.

16. Risks from Logging [State: Needs Investigation] [Priority: High]

Category: Tampering

Description: Log readers can come under attack via log files. Consider ways to canonicalize data in all logs. Implement a single reader for the logs, if possible, in order to reduce attack surface area. Be sure to understand and document log file elements which come from untrusted sources.

Justification: Determine what protections Elasticsearch offers for its log files. If an attacker can tamper with the log files, can the tampering go undetected?

17. Insufficient Auditing [State: Mitigation Implemented] [Priority: High]

Category: Repudiation

Description: Does the log capture enough data to understand what happened in the past? Do your logs

capture enough data to understand an incident after the fact? Is such capture lightweight enough to be left on all the time? Do you have enough data to deal with repudiation claims? Make sure you log sufficient and appropriate data to handle a repudiation claims. You might want to talk to an audit expert as well as a privacy expert about your choice of data.

Justification: Elasticsearch's audit log events capture enough security information to build timelines after the fact (See https://www.elastic.co/guide/en/elasticsearch/reference/current/audit-event-types.html).

18. Data Logs from an Unknown Source [State: Mitigation Implemented] [Priority: High]

Category: Repudiation

Description: Do you accept logs from unknown or weakly authenticated users or systems? Identify and authenticate the source of the logs before accepting them.

Justification: Access to the File System via Elasticsearch is controlled by Elasticsearch's user authentication. Any other access to the Elasticsearch files on each server will be mitigated using conventional role-based access control on the OS.

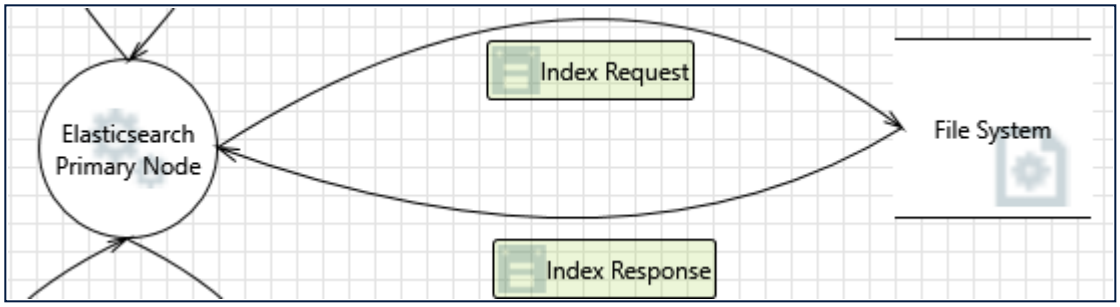
19. Authorization Bypass [State: Needs Investigation] [Priority: High]

Category: Information Disclosure

Description: Can you access File System and bypass the permissions for the object? For example by editing the files directly with a hex editor, or reaching it via filesharing? Ensure that your program is the only one that can access the data, and that all other subjects have to use your interface.

Justification: Verify that Elasticsearch does not expose raw File System access. If not, the conventional OS controlled access should be sufficient.

Interaction: Index Response



20. Risks from Logging [State: Needs Investigation] [Priority: High]

Category: Tampering

Description: Log readers can come under attack via log files. Consider ways to canonicalize data in all logs. Implement a single reader for the logs, if possible, in order to reduce attack surface area. Be sure to understand and document log file elements which come from untrusted sources.

Justification: Determine what protections Elasticsearch offers for its log files. If an attacker can tamper with the log files, can the tampering go undetected?

21. Weak Access Control for a Resource [State: Not Applicable] [Priority: High]

Category: Information Disclosure

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

Justification: The index files' paths are built into Elasticsearch. If an attacker has access to the File System, they have access to the machine that Elasticsearch is running on. This should be mitigated with OS role based access control outside of Elasticsearch.

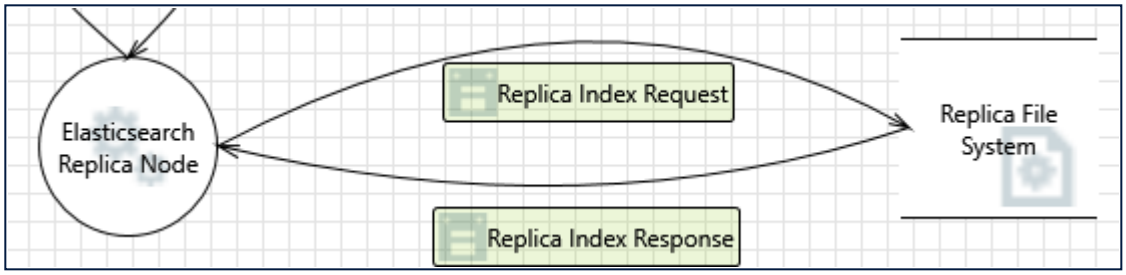
22. Spoofing of Source Data Store File System [State: Not Applicable] [Priority: High]

Category: Spoofing

Description: File System may be spoofed by an attacker and this may lead to incorrect data delivered to Elasticsearch Primary Node. Consider using a standard authentication mechanism to identify the source data store.

Justification: The index files' paths are built into Elasticsearch. If an attacker has access to the File System, they have access to the machine that Elasticsearch is running on. This should be mitigated with OS role based access control outside of Elasticsearch.

Interaction: Replica Index Request



23. Spoofing of Destination Data Store Replica File System [State: Not Applicable] [Priority: High]

Category: Spoofing

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

Justification: The index files' paths are built into Elasticsearch. If an attacker has access to the File System, they have access to the machine that Elasticsearch is running on. This should be mitigated with OS role based access control outside of Elasticsearch.

24. Risks from Logging [State: Needs Investigation] [Priority: High]

Category: Tampering

Description: Log readers can come under attack via log files. Consider ways to canonicalize data in all logs. Implement a single reader for the logs, if possible, in order to reduce attack surface area. Be sure to understand and document log file elements which come from untrusted sources.

Justification: Determine what protections Elasticsearch offers for its log files. If an attacker can tamper with the log files, can the tampering go undetected?

25. Lower Trusted Subject Updates Logs [State: Needs Investigation] [Priority: High]

Category: Repudiation

Description: If you have trust levels, is anyone other outside of the highest trust level allowed to log? Letting everyone write to your logs can lead to repudiation problems. Only allow trusted code to log.

Justification: Determine what protections Elasticsearch offers for its log files. If an attacker can tamper with the log files, can the tampering go undetected?

26. Data Logs from an Unknown Source [State: Mitigation Implemented] [Priority: High]

Category: Repudiation

Description: Do you accept logs from unknown or weakly authenticated users or systems? Identify and authenticate the source of the logs before accepting them.

Justification: Access to the File System via Elasticsearch is controlled by Elasticsearch's user authentication. Any other access to the Elasticsearch files on each server will be mitigated using conventional role-based access control on the OS.

27. Insufficient Auditing [State: Mitigation Implemented] [Priority: High]

Category: Repudiation

Description: Does the log capture enough data to understand what happened in the past? Do your logs capture enough data to understand an incident after the fact? Is such capture lightweight enough to be left on all the time? Do you have enough data to deal with repudiation claims? Make sure you log sufficient and appropriate data to handle a repudiation claims. You might want to talk to an audit expert as well as a privacy expert about your choice of data.

Justification: Elasticsearch's audit log events capture enough security information to build timelines after the fact (See https://www.elastic.co/guide/en/elasticsearch/reference/current/audit-event-types.html).

28. Potential Weak Protections for Audit Data [State: Mitigation Implemented] [Priority: High]

Category: Repudiation

Description: Consider what happens when the audit mechanism comes under attack, including attempts to destroy the logs, or attack log analysis programs. Ensure access to the log is through a

reference monitor, which controls read and write separately. Document what filters, if any, readers can rely on, or writers should expect

Justification: Access to the File System via Elasticsearch is controlled by Elasticsearch's user authentication. Any other access the Elasticsearch log files for auditing purposes on each server will be mitigated using conventional role-based access control on the OS.

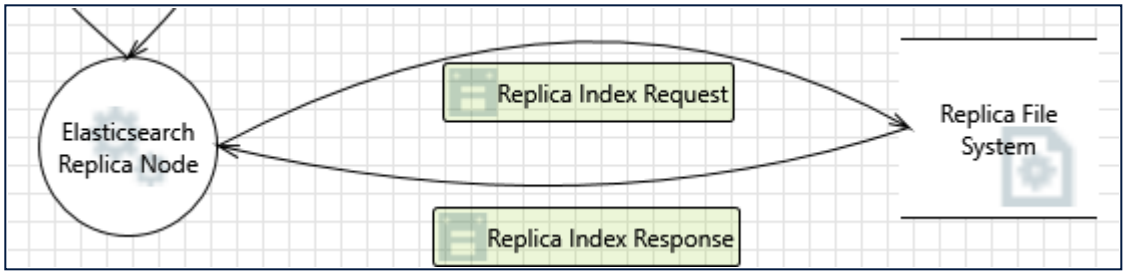
29. Potential Excessive Resource Consumption for Elasticsearch Replica Node or Replica File System [State: Needs Investigation] [Priority: High]

Category: Denial Of Service

Description: Does Elasticsearch Replica Node or Replica File System 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: Need to investigate if Elasticsearch has any bandwidth limiting for file system access.

Interaction: Replica Index Response



30. Spoofing of Source Data Store Replica File System [State: Not Applicable] [Priority: High]

Category: Spoofing

Description: Replica File System may be spoofed by an attacker and this may lead to incorrect data delivered to Elasticsearch Replica Node. Consider using a standard authentication mechanism to identify the source data store.

Justification: The index files' paths are built into Elasticsearch. If an attacker has access to the File System, they have access to the machine that Elasticsearch is running on. This should be mitigated with OS role based access control outside of Elasticsearch.

31. Risks from Logging [State: Needs Investigation] [Priority: High]

Category: Tampering

Description: Log readers can come under attack via log files. Consider ways to canonicalize data in all logs. Implement a single reader for the logs, if possible, in order to reduce attack surface area. Be sure to understand and document log file elements which come from untrusted sources.

Justification: Determine what protections Elasticsearch offers for its log files. If an attacker can tamper with

the log files, can the tampering go undetected?

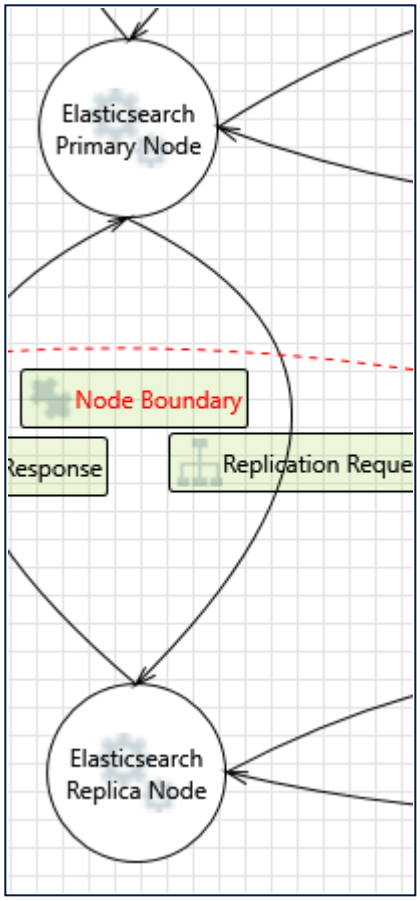
32. Weak Access Control for a Resource [State: Not Applicable] [Priority: High]

Category: Information Disclosure

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

Justification: The index files' paths are built into Elasticsearch. If an attacker has access to the File System, they have access to the machine that Elasticsearch is running on. This should be mitigated with OS role based access control outside of Elasticsearch.

Interaction: Replication Request



33. 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: Elasticsearch provides the option to include CSRF token protection.

34. Elevation by Changing the Execution Flow in Elasticsearch Replica Node [State: Needs Investigation] [Priority: High]

Category: Elevation Of Privilege

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

Justification: Further investigation is required to determine if program flow can be altered in harmful ways by new data.

35. Elasticsearch Replica Node May be Subject to Elevation of Privilege Using Remote Code Execution [State: Mitigation Implemented] [Priority: High]

Category: Elevation Of Privilege

Description: Elasticsearch Primary Node may be able to remotely execute code for Elasticsearch Replica Node.

Justification: Elatsicsearch's API does not allow for arbitrary remote code execution.

36. Elevation Using Impersonation [State: Not Applicable] [Priority: High]

Category: Elevation Of Privilege

Description: Elasticsearch Replica Node may be able to impersonate the context of Elasticsearch Primary Node in order to gain additional privilege.

Justification: If an attacker has access to a replica node, they likely have all or most of the data available on the primary node already.

37. Data Flow Replication Request Is Potentially Interrupted [State: Needs Investigation] [Priority: High]

Category: Denial Of Service

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

Justification: Need to investigate how Elasticsearch handles not being able to replicate data to other nodes

in the cluster.

38. Potential Process Crash or Stop for Elasticsearch Replica Node [State: Mitigation Implemented] [Priority: High]

Category: Denial Of Service

Description: Elasticsearch Replica Node crashes, halts, stops or runs slowly; in all cases violating an availability metric.

Justification: The Elasticsearch cluster is configured for resiliency. Configured with the recommending guidelines (See https://www.elastic.co/guide/en/elasticsearch/reference/current/high-availability-cluster-design.html), loss of the replica node can be handled.

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

Category: Information Disclosure

Description: Data flowing across Replication Request 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: SSL/TLS encryption in the Elasticsearch base will prevent unwanted information disclosure.

40. Potential Data Repudiation by Elasticsearch Replica Node [State: Mitigation Implemented] [Priority: High]

Category: Repudiation

Description: Elasticsearch Replica Node 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: Audit logging will be in place for all nodes, which is a feature of Elasticsearch and has to simply be enabled.

41. Potential Lack of Input Validation for Elasticsearch Replica Node [State: Mitigation Implemented] [Priority: High]

Category: Tampering

Description: Data flowing across Replication Request may be tampered with by an attacker. This may lead to a denial of service attack against Elasticsearch Replica Node or an elevation of privilege attack against Elasticsearch Replica Node or an information disclosure by Elasticsearch Replica Node. 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: Elasticsearch nodes can be configured to communicate using TLS/SSL (https://www.elastic.co/guide/en/elasticsearch/reference/current/ssl-tls.html) preventing

tampering.

42. Spoofing the Elasticsearch Replica Node Process [State: Needs Investigation] [Priority: High]

Category: Spoofing

Description: Elasticsearch Replica Node may be spoofed by an attacker and this may lead to information disclosure by Elasticsearch Primary Node. Consider using a standard authentication mechanism to identify the destination process.

Justification: Need to investigate what security mechanisms are in place in Elasticsearch when nodes are added (https://www.elastic.co/guide/en/elasticsearch/reference/current/modules-discovery-hosts-providers.html).

43. Weak Authentication Scheme [State: Mitigation Implemented] [Priority: High]

Category: Information Disclosure

Description: Custom authentication schemes are susceptible to common weaknesses such as weak credential change management, credential equivalence, easily guessable credentials, null credentials, downgrade authentication or a weak credential change management system. Consider the impact and potential mitigations for your custom authentication scheme.

Justification: Elasticsearch has a robust, extendable and customizable authentication scheme that can scale up in its strictness with relative ease.

44. Replay Attacks [State: Mitigation Implemented] [Priority: High]

Category: Tampering

Description: Packets or messages without sequence numbers or timestamps can be captured and replayed in a wide variety of ways. Implement or utilize an existing communication protocol that supports anti-replay techniques (investigate sequence numbers before timers) and strong integrity.

Justification: Elasticsearch provides Kerberos replay protection.

45. Collision Attacks [State: Needs Investigation] [Priority: High]

Category: Tampering

Description: Attackers who can send a series of packets or messages may be able to overlap data. For example, packet 1 may be 100 bytes starting at offset 0. Packet 2 may be 100 bytes starting at offset 25. Packet 2 will overwrite 75 bytes of packet 1. Ensure you reassemble data before filtering it, and ensure you explicitly handle these sorts of cases.

Justification: Need to investigate if Elasticsearch has protections for collision attacks.

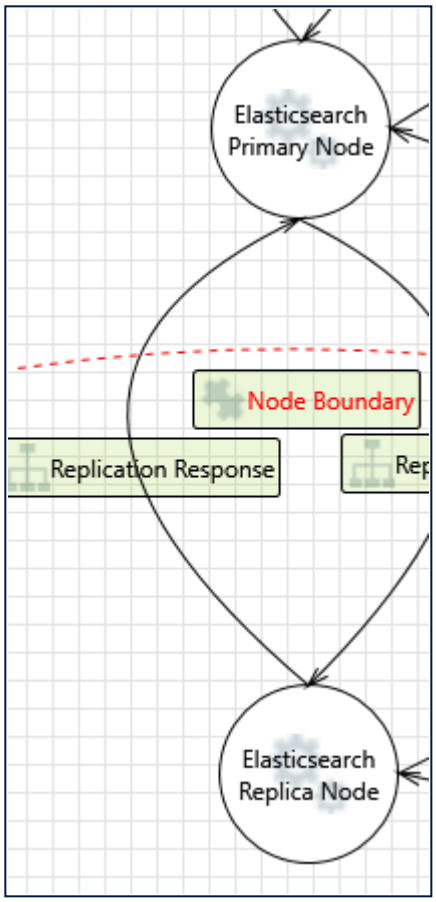
46. Elasticsearch Primary Node Process Memory Tampered [State: Not Applicable] [Priority: High]

Category: Tampering

Description: If Elasticsearch Primary Node is given access to memory, such as shared memory or pointers, or is given the ability to control what Elasticsearch Replica Node executes (for example, passing back a function pointer.), then Elasticsearch Primary Node can tamper with Elasticsearch Replica Node. Consider if the function could work with less access to memory, such as passing data rather than pointers. Copy in data provided, and then validate it.

Justification: If the Primary Node is compromised, then the attacker already has access to customer account information. This should be mitigated with standard OS role based access control to the machines that both Primary and Replica node live on.

Interaction: Replication Response



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

Category: Information Disclosure

Description: Data flowing across Replication Response 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: Our Elasticsearch base would be built on SSL/TLS encryption, so this would mitigate information disclosure.

48. Elevation Using Impersonation [State: Not Applicable] [Priority: High]

Category: Elevation Of Privilege

Description: Elasticsearch Primary Node may be able to impersonate the context of Elasticsearch Replica Node in order to gain additional privilege.

Justification: If the attacker already has access to the Elasticsearch Primary Node, they have access to all the data already.

49. 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: Elasticsearch provides the option to include CSRF token protection.

50. Elevation by Changing the Execution Flow in Elasticsearch Primary Node [State: Needs Investigation] [Priority: High]

Category: Elevation Of Privilege

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

Justification: Further investigation is required to determine if program flow can be altered in harmful ways by new data.

51. Potential Data Repudiation by Elasticsearch Primary Node [State: Mitigation Implemented] [Priority: High]

Category: Repudiation

Description: Elasticsearch Primary Node 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: Elasticsearch is equipped with an audit logging function that simply must be enabled and will record this data.

52. Potential Lack of Input Validation for Elasticsearch Primary Node [State: Mitigation Implemented] [Priority: High]

Category: Tampering

Description: Data flowing across Replication Response may be tampered with by an attacker. This may lead to a denial of service attack against Elasticsearch Primary Node or an elevation of privilege attack against Elasticsearch Primary Node or an information disclosure by Elasticsearch Primary Node. 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: Elasticsearch nodes can be configured to communicate using TLS/SSL (https://www.elastic.co/guide/en/elasticsearch/reference/current/ssl-tls.html) preventing tampering.

53. Spoofing the Elasticsearch Primary Node Process [State: Needs Investigation] [Priority: High]

Category: Spoofing

Description: Elasticsearch Primary Node may be spoofed by an attacker and this may lead to information disclosure by Elasticsearch Replica Node. Consider using a standard authentication mechanism to identify the destination process.

Justification: Need to investigate what security mechanisms are in place in Elasticsearch for nodes validating/authenticating other nodes (https://www.elastic.co/guide/en/elasticsearch/reference/current/modules-discovery-hosts-providers.html).

54. Elasticsearch Primary Node May be Subject to Elevation of Privilege Using Remote Code Execution [State: Not Applicable] [Priority: High]

Category: Elevation Of Privilege

Description: Elasticsearch Replica Node may be able to remotely execute code for Elasticsearch Primary Node.

Justification: Elasticsearch's API does not allow for arbitrary remote code execution.

55. Data Flow Replication Response 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: Elasticsearch implements several replica nodes which mitigates damage done by any one node being interrupted.

56. Potential Process Crash or Stop for Elasticsearch Primary Node [State: Mitigation Implemented] [Priority: High]

Category: Denial Of Service

Description: Elasticsearch Primary Node crashes, halts, stops or runs slowly; in all cases violating an availability metric.

Justification: The data on the primary node is available on the replica nodes, and the clusters make sure there is always at least one node active.

57. Elasticsearch Replica Node Process Memory Tampered [State: Not Applicable] [Priority: High]

Category: Tampering

Description: If Elasticsearch Replica Node is given access to memory, such as shared memory or pointers, or is given the ability to control what Elasticsearch Primary Node executes (for example, passing back a function pointer.), then Elasticsearch Replica Node can tamper with Elasticsearch Primary Node. Consider if the function could work with less access to memory, such as passing data rather than pointers. Copy in data provided, and then validate it.

Justification: If the Primary Node is compromised, then the attacker already has access to customer account information. This should be mitigated with standard OS role based access control to the machines that both Primary and Replica node live on.

58. Replay Attacks [State: Mitigation Implemented] [Priority: High]

Category: Tampering

Description: Packets or messages without sequence numbers or timestamps can be captured and replayed in a wide variety of ways. Implement or utilize an existing communication protocol that supports anti-replay techniques (investigate sequence numbers before timers) and strong integrity.

Justification: Elasticsearch provides Kerberos replay protection.

59. Collision Attacks [State: Needs Investigation] [Priority: High]

Category: Tampering

Description: Attackers who can send a series of packets or messages may be able to overlap data. For example, packet 1 may be 100 bytes starting at offset 0. Packet 2 may be 100 bytes starting at offset 25. Packet 2 will overwrite 75 bytes of packet 1. Ensure you reassemble data before filtering it, and ensure you explicitly handle these sorts of cases.

Justification: Need to investigate if Elasticsearch has protections for collision attacks.

60. Spoofing the Elasticsearch Replica Node Process [State: Needs Investigation] [Priority: High]

Category: Spoofing

Description: Elasticsearch Replica Node may be spoofed by an attacker and this may lead to unauthorized access to Elasticsearch Primary Node. Consider using a standard authentication mechanism to identify the source process.

Justification: Need to investigate what security mechanisms are in place in Elasticsearch for nodes validating/authenticating other nodes (https://www.elastic.co/guide/en/elasticsearch/reference/current/modules-discovery-hosts-providers.html).