NetWitness® Platform XDR

Version 12.1.0.0

Data Privacy Management Guide

NETWITNESS
Platform XDR

Contact Information

NetWitness Community at https://community.netwitness.com contains a knowledge base that answers common questions and provides solutions to known problems, product documentation, community discussions, and case management.

Trademarks

License Agreement

Third-Party Licenses

Note on Encryption Technologies

Distribution

October, 2022

Contents

Data Privacy Overview

This topic introduces the concept and implementation considerations for a data privacy officer or administrator who is managing exposure of privacy-sensitive data in NetWitness. In addition, information about recommended use cases is included.

Note: A data privacy plan touches on most components of NetWitness. The person who configures data privacy needs to understand NetWitness network components, configuration of NetWitness hosts and services as described in the *Host and Services Getting Started Guide*, and the types of information that need to be protected.

Regulatory mandates in some locations, for example the European Union (EU), require that information systems have a means of protecting privacy-sensitive data. Any data that could directly or indirectly identify "Who did what when?" may be considered privacy-sensitive data. A few examples are user names, email addresses, and host names. NetWitness provides a range of controls that customers can leverage to protect privacy-sensitive data. These controls can be used in a variety of combinations to protect privacy-sensitive data, without significantly reducing analytical capability.

A user role for a Data Privacy Officer (DPO) supports the management of privacy-sensitive data. The DPO can configure NetWitness to limit exposure of metadata and raw content (packets and logs) using a combination of techniques. The methods available to protect data in NetWitness include:

- Data Obfuscation
- Data Retention Enforcement
- Audit Logging

Data Obfuscation

NetWitness has configurable options for data obfuscation. Data privacy officers and administrators can specify which meta keys in their environment are privacy-sensitive and limit where the meta values and raw data for those keys are displayed in the NetWitness network. In place of the original values, NetWitness can provide obfuscated representations to enable investigation and analytics. In addition, DPOs and administrators can prevent persistence of privacy-sensitive meta values and raw logs or packets.

Three methods work together to implement data obfuscation:

- Obfuscation of meta values for privacy-sensitive meta keys with an optional salt. Meta keys configured as protected are represented by obfuscated values at the time of creation on a Decoder or Log Decoder; the obfuscated values are hashed and considered to be impossible to read. To implement, you must configure the Decoder and Log Decoder hash algorithm and salt, and configure privacy-sensitive language keys as protected on all Core services.
- Role-based access (RBAC) to the raw logs or packets and the privacy-sensitive meta values. The DPO can use roles with granular permission capabilities to restrict what an analyst versus a data privacy officer is able to view during configuration, analysis, and investigation. The *System Security and User Management Guide* provides in-depth coverage of the RBAC implementation in NetWitness. To implement, you must configure metadata and content visibility per role on individual Brokers, Concentrators, Decoders, Log Decoders, and Archivers.

- Preventing persistence of privacy-sensitive meta values and raw logs or packets. To implement, you must configure meta keys on parsers for individual Decoders and Log Decoders as transient.

Data Retention Enforcement

NetWitness can ensure that data is retained only as long as necessary or as specified. An administrator can configure data retention using age and time thresholds on a per-service basis. Schedulers running on each service automatically delete data meeting those thresholds. Once the data is deleted, it is no longer available through user interfaces, queries, or application programming interface (API) calls. Some of the NetWitness components also support purging of data through overwrites.

An administrator can manage data retention in several ways:

- Configure how long data persists in storage on the system.
- For Core services, strategically remove privacy-sensitive data that may have been written by configuring automatic removal of data of a specific age.
- Configure NetWitness so that original data is not sent or saved to the other components. If privacy-sensitive data makes its way into another database on the Reporting Engine, Malware Analysis, and NetWitness Servers, data retention can be managed there as well. This configuration for Event Stream Analysis is managed in the Services Explorer view.

Note: If a situation arises where the DPO decides that already collected data is privacy-sensitive after the system is functional, the administrator can manually overwrite the data from databases or files where the data is saved.

Audit Logging

Administrators can leverage audit logs that NetWitness creates using the Global Audit Logging feature. The audit logging feature generates audit log entries about many activities, and the following are examples of log entries that are relevant to data privacy:

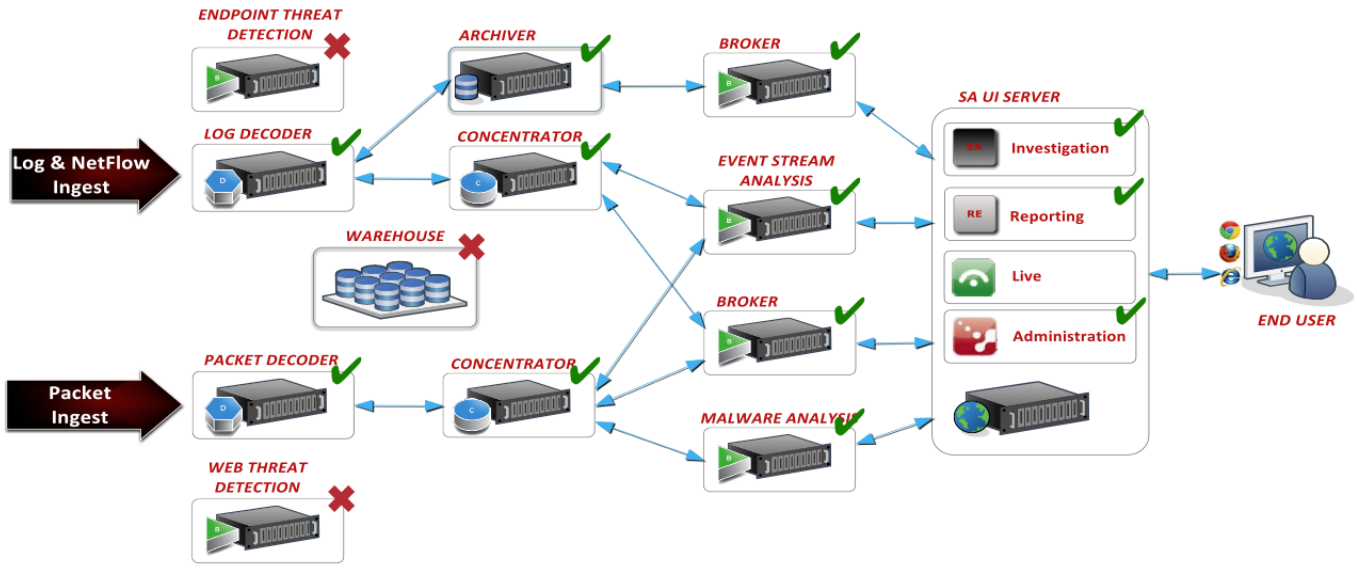
- Modifications to permissions and users assigned to roles.
- Failed and successful attempts to log on to NetWitness and log off.
- Data deletion.
- Data exports and downloads.
- Navigation by users to user interfaces and queries that users performed.
- Attempts (successful or not) to view or modify privacy-sensitive data, including an identification of the user who made the attempts.

All audit log entries are part of a standard audit trail for NetWitness. Administrators can configure NetWitness to forward audit logs to a designated destination, including third-party systems to provide additional filtering and reporting capabilities. For more information on Global Audit Logging, see "Configure Global Audit Logging" in the *System Configuration Guide*.

Components Covered by the Data Privacy Feature

The figure below identifies the NetWitness components covered by the Version 11.x data privacy feature with a green check mark. Components marked with an X are not supported by data privacy functions. The *NetWitness Getting Started Guide* provides a functional description of NetWitness components.

Note: NetWitness data privacy features are not supported for Warehouse and protected metadata can make it to Warehouse through Warehouse Connector, unless explicitly configured to be filtered out using Warehouse Connector Meta Filters. If protected metadata makes it to Warehouse, users having direct access to Warehouse can query such data. Data privacy officers need to prevent that through administrative, technical, and procedural controls outside of NetWitness.



Data Privacy Feature Implementation by Component

The following table identifies which data privacy features are supported for each NetWitness component. For each component, a check mark indicates if the component supports data obfuscation, data retention enforcement, data overwriting, and audit logging.

Component	Data Obfuscation	Data Retention Enforcement	Data Overwriting	Audit Logging
Ingestion				
Decoder	✓	✓	✓	✓
Log Decoder	✓	✓	✓	✓
Meta Aggregation				
Concentrator	✓	✓	✓	✓

Component	Data Obfuscation	Data Retention Enforcement	Data Overwriting	Audit Logging
Broker	n/a	✓ (stored in DPO cache only)[1]		✓
Real-Time Analysis				
Investigate	✓	✓ (stored in DPO cache only)[2]		✓
Event Stream Analysis	✓			✓
Malware Analysis	✓	✓		✓
Respond	✓	✓		✓
Reporting				
Reporting Engine	✓	✓		✓
Long-Term Analytics				
Archiver	✓	✓	✓(uncompressed)[3]	✓
Warehouse				

Notes:
1 - Brokers can cache data and this needs to be cleared by configuring an independent rollover and other removal of cache as required. The administrator can configure cache rollover for a Broker using the Scheduler in the Services Config view Files tab.
2 - Investigate and the NetWitness Server cache data, and this is cleared automatically every 24 hours.
3 - The overwriting procedure described in Configure Data Retention applies to uncompressed data.

Component-Specific Configuration Guidelines

NetWitness components and modules that obtain access to privacy-sensitive metadata and their obfuscated counterparts are Investigate, Event Stream Analysis (ESA), Malware Analysis, Respond, and Reports. They get access to data based on the permissions defined for the role to which the user belongs. The Administrator or DPO configures each Decoder or Log Decoder to identify meta keys that are flagged for obfuscation.

These components have additional guidelines to ensure that they function as expected with a data privacy scheme:

- **Event Stream Analysis**. When ESA receives privacy-sensitive data from NetWitness core, ESA passes on only the obfuscated version of the data. ESA does not store or show protected data. There are some additional guidelines for configuring advanced EPL rules and enrichment sources as well as information on how to remove sensitive data globally from all alerts (see "How ESA Handles Sensitive Data" in the *Alerting with ESA Correlation Rules User Guide*).

- **Malware Analysis**. Malware Analysis references certain meta keys during scoring, including `alias.host`, `client`, and others. To ensure no loss of analytical functionality Malware Analysis should be configured as a trusted client; that is, configured to connect to the NetWitness Core infrastructure with an account equivalent to a user in DPO role. Otherwise, if meta keys referenced by Malware Analysis do get tagged for obfuscation and are not accessible to Malware Analysis, some of the Indicators of Compromise (IOCs) may be rendered ineffective.
- **Respond Server service.** The Respond Server service uses a data privacy mapping file to display obfuscated data in alerts.(see "Obfuscate Private Data" in the *NetWitness Respond User Guide*) and has a configurable data retention period for alerts (see "Set a Retention Period for Alerts and Incidents" in the *NetWitness Respond User Guide*).
- **Reports.** In Reporting Engine, each Core service is added as two separate data sources, using the two separate service accounts; one data source has a service account representing the Data Privacy Officer role and the other data source has a service account representing a non-Data Privacy Officer role. "Configure Data Privacy for Reporting Engine" in the *Reporting Engine Configuration Guide* provides procedures to configure data privacy for Reporting Engine.

Recommended Configurations

This topic describes the recommended data privacy implementation for NetWitness and several additional use cases for managing exposure of privacy-sensitive data in NetWitness. Administrators can set up the NetWitness hosts and services to meet data privacy requirements for their environment. This section provides recommended configurations for data privacy and data retention.

Recommended Data Privacy Configuration

The recommended configuration to obtain the best analytical value with data obfuscation enabled is to define privacy-sensitive metadata and keep both original and obfuscated (hash) values of privacy-sensitive data on disk for Decoders, Log Decoders, Concentrators, and Brokers.

The assumption is that only a handful of metadata (approximately 10 meta keys) will be classified as protected and a FIPS 140-compliant algorithm for hashing will be used along with a salt to make reverse engineering the original value difficult. The recommended solution is SHA-256 with a salt length of at least 16 characters and up to 60 characters.

Note: By default, hash values are stored in binary format for faster response times and because it requires less storage space in the database when compared to saving them in string format. The recommended storage method is text/string.

Brokers and Investigate may have original and obfuscated data in cache due to data privacy officers using Investigate to confirm the original value to which the obfuscated value maps during investigations. Downstream services can also limit the use of the original sensitive values to in-memory processing so that data does not persist on disk in those downstream systems; this holds true for ESA and Malware Analysis.

The recommended solution to delete data when ready is the built-in and automatic data retention enforcement, which deletes data at a certain threshold. You can use this method for the following components: Decoder, Log Decoder, Log Collector, Archiver, Malware Analysis, NetWitness Respond, and Reporting Engine. You can manually configure Event Stream Analysis to support similar automatic data retention enforcement.

To manage cache storage, the NetWitness Server clears cache related to investigations of events every 24 hours. The Broker can also be configured to execute a periodic removal of locally stored cache.

Options for Data Retention Configurations

NetWitness provides alternative controls that the administrator can apply to enforce stronger restrictions on privacy-sensitive data storage when data obfuscation is enabled.

Data Storage With Data Retention Options in Effect

The following table summarizes where data is stored in the default configuration with no data privacy as well as for each data retention alternative. A checkmark indicates that privacy-sensitive data is saved on the component; a blank indicates that no privacy-sensitive data is stored on the component.

Component	Default Configuration	Data Storage Options		
	Original Data Stored	Original Data and Hash Stored (recommended)	Only Hash Stored	No Data Stored (all metadata is transient)
Ingestion				
Decoder	✓	✓		
Log Decoder	✓	✓		
Meta Aggregation				
Concentrator	✓	✓		
Broker	✓ (Cache only)	✓ (Cache only)		
Real-Time Analysis				
Investigate	✓	✓ (Cache only)		
Event Stream Analysis	✓			
Malware Analysis	✓			
Respond	✓			
Reporting				
Reporting Engine	✓	✓ (Optional)		
Long-Term Analytics				
Archiver	✓ (Optional)	✓ (Optional)		
Warehouse	✓ (Optional)	✓ (Optional)		
Content				
Live	n/a	n/a	n/a	n/a
Fraud Analysis				
RSA Fraud and Risk Intelligence Suite	n/a	n/a	n/a	n/a

Component	Default Configuration	Data Storage Options		
	Original Data Stored	Original Data and Hash Stored (recommended)	Only Hash Stored	No Data Stored (all metadata is transient)
End Point Protection				
NetWitness Endpoint	n/a	n/a	n/a	n/a

Notes:
Cache Only means that sensitive data is in the Broker or NetWitness Server cache. Configure Data Retention provides details about automated and manual clearing of cache.
Optional means that sensitive data storage does occur, but can be limited by optional configurations. For example, to limit where sensitive data is stored, do not enable DPO access for Reporting and do not aggregate original protected data into the Archiver.

Option 1: No Original Data Saved to Disk, Only Hash Stored

Administrators can eliminate the persistence of sensitive data to disk and store only an obfuscated value if the risk of exposure is too great. In this scenario, metadata generated during parsing on the Decoders and Log Decoders is used only in memory and not written to disk. Administrators can configure individual meta keys on a Decoder or Log Decoder as transient to ensure that sensitive metadata is not written to disk. Downstream services do not see original values and must use obfuscated values to conduct investigation and analytics.

To configure this data privacy scheme, data obfuscation must be enabled with hash values configured. You can configure individual meta keys on a Decoder or Log Decoder as transient to ensure that original values are not written to disk.

- Original values identified as sensitive are extracted from the raw data during parsing on the Decoder and Log Decoder and are accessible to the system during parsing (parsers, rules, feeds).
- The Decoder does not save the original values for meta keys identified as sensitive, storing only the hash of original values along with other non-sensitive metadata related to the event.

A side effect of these options is some loss in analytical capability, but you can configure these to suit the needs of your environment.

- By configuring all sensitive data as Transient, sensitive values are not persisted to disk, and the analytic capabilities using the original value are available at parse time only (parsers, rules, feeds).
- Event stream analysis (ESA) and Malware Analysis systems must rely only on the obfuscated meta values when doing their correlation and scoring respectively.
- Reporting Engine is limited to pulling reports using the non-sensitive and obfuscated values.

- The data privacy officer cannot view the original value, but can use the configured hash and salt to determine if an obfuscated value represents a specific known original value.

Option 2: No Original or Obfuscated Values Stored: Not Recommended

Administrators can eliminate the persistence of the original value to disk entirely if the risk of exposure is too great. As in Option 1, in this scenario, metadata generated during parsing on the Decoders and Log Decoders is used only in memory and not written to disk. Administrators can configure individual meta keys on a Decoder or Log Decoder as transient to ensure that sensitive metadata is not written to disk. Downstream services do not see original values and have no obfuscated values to conduct investigation and analytics.

To configure this data privacy scheme, configure individual meta keys on a Decoder or Log Decoder as transient to ensure that original values are not written to disk.

- Original values identified as sensitive are extracted from the raw data during parsing on the Decoder and Log Decoder and are accessible to the system during parsing (parsers, rules, feeds).
- The Decoder does not save not save the original values for meta keys identified as sensitive, storing only non-sensitive metadata related to the event.

A side effect of these options is significant loss in analytical capability, but you can configure these to suit the needs of your environment.

- By configuring all sensitive data as Transient, sensitive values are not persisted to disk, and the analytic capabilities using the original value are available at parse time only (parsers, rules, feeds). See Configure Data Retention.
- All downstream components have no visibility in the original values, obfuscated or otherwise.
- The data privacy officer has no visibility into the original value obfuscated or otherwise.

Optional Data Overwriting Options

Several options for overwriting data are available, and you should thoroughly understand each one before implementing data overwriting.

Option 1: Limit Disk Space for Continuous Overwriting of Older Data

If the desired data retention period to store the data, and therefore the amount of storage required for that data, is known, the size of the underlying hardware or the partition can be limited to that size. By reducing the hard drive storage or the partition size, the amount of free space available that has to be filled before new data overwrites it would also be limited. The newly ingested data continually overwrites the older data. Either solution must be done at deployment time to be effective.

Side effects of this option are:

- The removal of some disks will limit the number of resources available to distribute the I/O, causing some degradation in performance.

- The smaller partition size may cause some degradation in performance, but would alleviate some of the performance impact of removing disks.

Option 2: Use Tiered Storage to Overwrite Data on a Scheduled Basis

If overwriting of data is required on a scheduled automatic basis, you can configure the Decoders and Concentrators to use tiered storage. The tiered storage configuration provides a mechanism for invoking a script after a database file has been removed from the application but prior to its removal from the file system. If necessary, instead of moving the file to the second tier, or cold storage, (the intended function in a tiered storage use case), the script can use a utility like the CentOS `shred` utility to overwrite the file. This tool is less effective when the database is stored in a journaling file system like XFS, in which the Core database resides, and on a RAID logical drive like the ones with which the Core hosts connect.

Most other NetWitness components do not have this option; their data is stored in a database that does not support the tiered storage mechanism. The only other component that could use this overwrite method is the Reporting Engine since it saves reports and alerts as individual files. However, the Reporting Engine charts are stored in a database so they would be immune to this technique.

Option 3: Purge Data Using String and Pattern Redaction Option

Data purging provides a mechanism to strategically overwrite a specific subset of data from the system in case any sensitive data has been persisted either on purpose or by accident. The NetWitness `wipe` utility allows for unique patterns to be written over the data in the meta and packet databases for Core services, which may contain RAW packets or logs for existing sessions, based on a session identifier. All Core components have the capability to overwrite a subset of data that has been found by executing a query string, including regex patterns. The session identifiers resulting from the query are fed into the NetWitness `wipe` utility.

Note: This option is not available if the data in the Core database has been compressed (as typically done in Archiver deployments).

In most NetWitness components the database in use does not provide a built-in redaction or secure deletion mechanism. The Malware Analysis component can overwrite the data object in the database with the value `private` instead of deleting it during the data retention management process, but this is not meant to be a secure deletion mechanism.

Caution: Using this method on a large number of sessions has two drawbacks: it can be time-consuming and impact performance.

See Purge Data Using String and Pattern Redaction Option for procedure.

Limitations to Data Overwriting

There are limitations to the overwriting techniques described as Option 2 and 3. To perform the overwrite of the data in the disk sectors, the above options for overwriting and the overwrite command line tool provided as an alternative method (`shred`, a function of CentOS) make assumptions about the disk layout. NetWitness hosts use SSD drives and RAID configurations for performance and reliability reasons, and these inhibit the functionality of the overwrite techniques. If overwrite techniques alter SSD drives and RAID configurations in an attempt to increase security, there will inevitably be an associated performance cost reflected in ingest rates, query speeds, and potentially other areas. The command line tools available for overwrite are recommended only for special use cases when it is necessary to redact specific data. The tools are not for use in a real-time continuous method because of the potential performance cost that will be incurred.

Quick Start Procedures

This section provides end-to-end instructions for preparing to configure data privacy features, then completing the configuration of the recommended data privacy solution.

- Prepare to Configure Data Privacy
- Configure the Recommended Data Privacy Solution

Prepare to Configure Data Privacy

This topic provides general guidelines for planning and configuring data privacy policies in the NetWitness network. Before beginning configuration, you must understand the data that needs to be protected on your network and develop a plan. You will need to:

1. Identify the meta keys that hold privacy-sensitive data and need to be protected. This decision is based on requirements specific to your site.
2. Decide which users need access to privacy-sensitive metadata and raw content. The first decision is whether to separate the DPO and administrator roles for your site by configuring a custom administrators system role on Decoder and Log Decoders and removing the `dpo.manage` permission. By default, administrators have all permissions including the ability to configure the salted hash transform used to obfuscate data; some sites may want to reserve this access for data privacy officers. "Service User Roles and Permissions" in the *Hosts and Services Getting Started Guide* has more details on exactly what permissions each role has and the purpose of the permissions.
3. Plan the configuration changes you need to make in your NetWitness deployment to support adequate data privacy.
4. Assess how your configuration may impact out-of-the-box and custom content. For example, by default content available through Live for Reporting Engine is not geared toward obfuscated meta values.

In a single deployment, certain data-privacy configurations in the Core services must be the same. The following table lists these settings and uses a checkmark to identify the services for which the configuration must be the same. In the heading, the following abbreviations are used: D = Decoder, LD = Log Decoder, A = Archiver, C = Concentrator, and B = Broker.

	Configuration must be the same for these services:				
Setting	**D**	**LD**	**A**	**C**	**B**
Hash algorithm and salt for privacy-sensitive data	✓	✓			
Language key data privacy attributes in the custom index file (includes configuring keys as protected)	✓	✓	✓	✓	✓
Transient meta keys (not persisted on disk) per service and parser	✓	✓			
Meta data and raw content visibility per system user group. (The meta keys must exist in the custom index file.)	✓	✓	✓	✓	✓
User who has the `Aggregation` service user role assigned is added.*	✓	✓	✓		

* When trying to access data on an aggregate service, the Log Collector or Broker requests authentication. When prompted to enter user name and password, you must authenticate as a user who is assigned the `Aggregation` service role. "Services Security View - Aggregation Role" in the *Hosts and Services Getting Started Guide* provides detailed information about this role. Follow the instructions in "Add, Replicate or Delete a Service User" in the *Hosts and Services Getting Started Guide* to create a user and assign the new user the Aggregation service user role.

Configure the Recommended Data Privacy Solution

This topic tells administrators and data privacy officers how to configure the recommended data privacy solution in a NetWitness network. These are the basic steps to follow to configure the NetWitness system to identify sensitive data and determine who can see the sensitive data. The recommended configuration generates obfuscated values of certain original meta keys and then persists both the original and obfuscated data so that it is available to users assigned privileged role access.

This configuration has several parts:

1. Create two users with different levels of permissions. One user (the data privacy officer) can view all metadata and another user (an analyst) is restricted from seeing certain metadata and content with associated metadata.
2. Set up two transforms using a salt and hash to create an obfuscated version of original `username` and `ip.src` meta keys.
3. Configure data retention on the Decoder and Concentrator services.

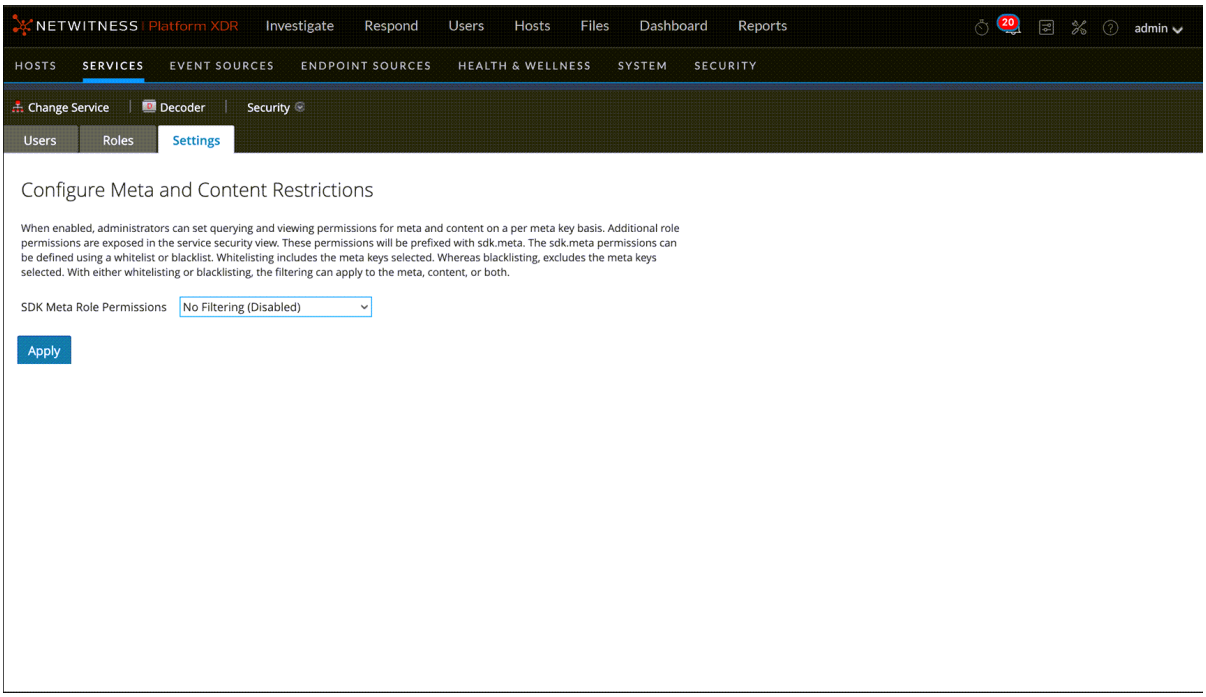
Note: The following conditions are required in order to complete this procedure:
-The Concentrator and Decoder must be added to the NetWitness Server using trusted connections.
-The NW Server version must be 11.x or later.
-The Core services must be 11.x or later.
-Aggregation must use Aggregators accounts on all Core services.

Configure Metadata and Content Restrictions on Brokers, Concentrators, and Decoders

Note: In version 11.6, if these 12 metas namely `'sessionid','nwe.callback_id', 'medium','session.split','ip.dst','ip.src','ipv6.src','ipv6.dst','tcp.dstport','tcp.srcport','udp.dstport', and 'udp.srcport'` are restricted then the group events option will be disabled.

To restrict the metadata and raw content that users can view, you must enable SDK system roles to allow more granular controls by configuring metadata and content restrictions on each service in the Services Security view.

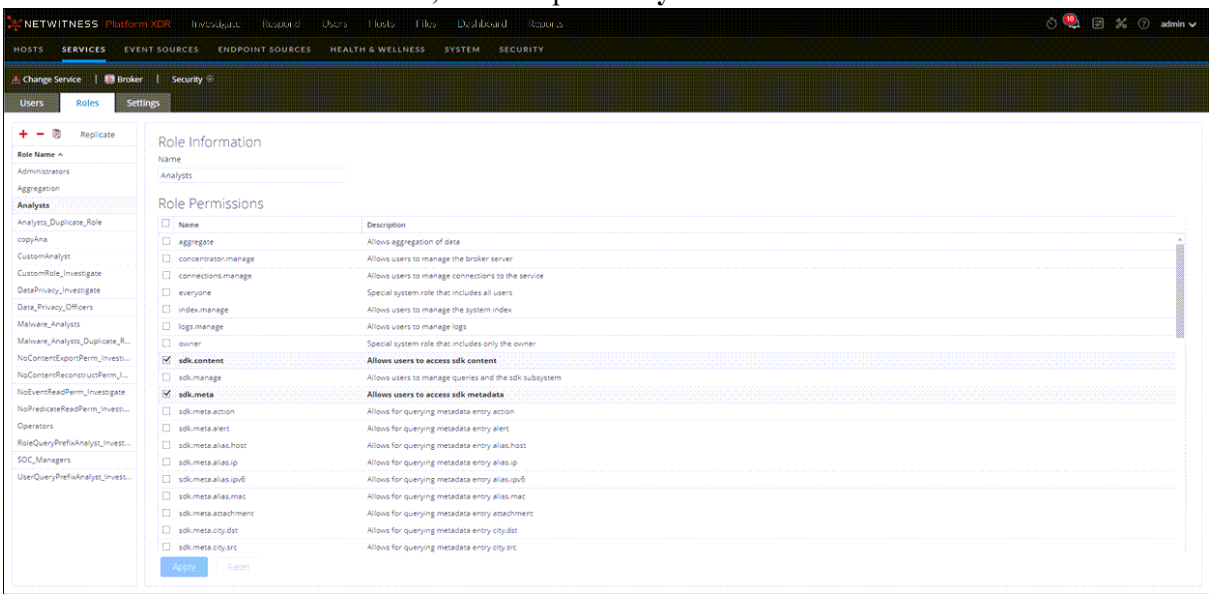
1. Go to  **(Admin)** > **Services**, select a service and then select  **> View > Security**.
2. Click the **Settings** tab.



3. In the **SDK Meta Role Permissions** field, select **Blacklist meta and content**. Click **Apply**.

   This allows the administrator to blacklist individual meta keys so that only the data privacy officer can see the meta keys and content. New roles per meta key are added to the Roles tab.

4. Click the **Roles** tab and select a role, for example **Analysts**.



5. In the **Roles** tab,

   a. Select the meta keys that you do not want analysts to see, for example, select `sdk.meta.username` and `sdk.meta.ip.src`.
   This restricts the analyst from seeing the privacy-sensitive meta keys `username` and `ip.src` as well as any content for any session that contains that metadata within it.

b. Ensure that `sdk.packets` is selected.
   If it is de-selected, analysts lose the ability to bulk export raw packets and logs. In NetWitness Platform 11.0 and later, RBAC just works for packets. Sessions that are restricted are just skipped during pcap generation in Investigate. Sessions that are allowed have packets returned. For more information on RBAC, see the *System Security and User Management Guide*.

c. Click **Apply**.

6. In the Roles tab, ensure that the `Data_Privacy_Officers` role has no `sdk.meta.values` selected. Click **Apply**.

   A DPO can view any metadata and any session.

   In the Roles tab, ensure that the `Aggregation` role has the following permissions: select `aggregate`, `sdk.content`, `sdk.meta`, and `sdk.packets`.

Add Data Privacy Officer and Analyst Accounts on the NetWitness Server

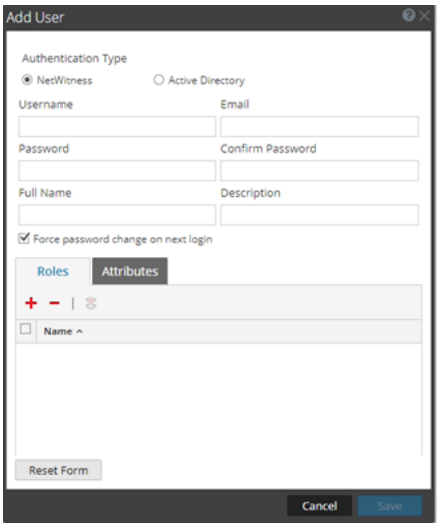
You must add two new user accounts in NetWitness at the system level to depict a privileged data privacy officer and a typical analyst. If the environment is configured using the default trusted connections, you do not need to create the new user accounts on the Core services (Brokers, Concentrators, and Decoders). When a user is created in the NetWitness Server, that user can log on to the services.

Note: The role name is required to exist on both the server and the services, and the role name for all must be identical. If you create a new custom role on the NetWitness Server, make sure to add it to all Core services as well.

1. Create a new user account for the data privacy officer:

   a. Go to **(Admin)** > **Security**, select the **Users** tab. In the **Users** tab toolbar, click .

   The Add User dialog is displayed.



   b. Create the new account with the following credentials.

Username = <new user name for logon, for example, DPOadmin>
Email = <new user's email, for example, DPOadmin@rsa.com>
Password = <new user's password for logging on, for example, RSAprivacy1!@>
Full Name = <new user's full name, for example, DPO Administrator>

c. Click the Roles tab,  , and select the `Data_Privacy_Officers` role for the new user.

d. Select **Save**.

2. Create a new user account for the analyst with limited privileges:

a. In the **Services Security** view, select the **Users** tab. In the **Users** tab toolbar, click  .

The Add User dialog is displayed.

b. Create the new account with the following credentials:

Username = <new user name for logon, for example, NonprivAnalyst>
Email = <new user's email, for example, NonprivAnalyst@rsa.com>
Password = <new user's password for logging on, for example, RSAprivacy2!@>
Full Name = <new user's full name, for example, Nonprivileged Analyst>

c. Click the Roles tab,  , and select the `Analysts` role for the new user.

d. Select **Save**.

Configure Obfuscated Data on Decoders and Concentrators

This procedure creates the obfuscated values to provide to users who do not have access to the original values.

1. Configure a salt so that the obfuscated value becomes unique. Different companies may have analysts of the same first name and potentially the same login username, and using a salt limits the possibility of someone outside your organization determining your obfuscation mechanism. In this example, you use a simple salt and SHA-256, but the salt is configurable and the hash algorithm can be changed. For additional information, see Configure Data Obfuscation.

a. To define the salt and hash algorithm, go to  **(Admin) > Services**.

b. Select a Decoder in the **Admin Services** view and select  > **View > Config**.

c. Click the **Data Privacy** tab, and select hash algorithm (SHA-256). In the Salt field, type a hash, for example, **rsasecurity**, and click **Apply**.

2. Define the transforms, including the hash format, between the original meta key and obfuscated meta key on the Decoder. The default hash format is binary, but the recommended configuration calls for using the text/string format.

a. While still in the Services Config view, click the **Files** tab, and in the drop-down menu select **index-decoder-custom.xml.** (You can apply this same configuration to the Log Decoder in the `index-logdecoder-custom.xml` file.)

b. Enter the following lines in the available input area:

```
<?xml version="1.0" encoding="utf-8"?>
<language level="IndexNone" defaultAction="Auto">
<key name="username" description="Username" format="Text"
protected="true"><transform destination="username.hash"/></key>
<key name="username.hash" description="Username Hash"
format="Text"/>
<key name="ip.src" description="Source IP Address" format="IPv4"
protected="true"><transform destination="ip.src.hash"/></key>
<key name="ip.src.hash" description="Source IP Address Hash"
format="Text"/>
</language>
```

c. To restart the Decoder service, go to **(Admin)** > **Services**, find the Decoder service you want to restart, and select > **Restart**.
The service should automatically restart.

3. Define the meta keys on the Concentrator in the `index-concentrator-custom.xml` file:

a. Go to **(Admin)** > **Services**, select a Concentrator in the **Admin Services** view, and select > **View > Config**.

b. Click the **Files** tab, and in the drop-down menu select **index-concentrator-custom.xml**

c. Enter the following lines in the available input area:

```
<?xml version="1.0" encoding="utf-8"?>
<language level="IndexValues" defaultAction="Auto">
<key name="username" description="Username" format="Text"
level="IndexValues" protected="true"/>
<key name="username.hash" description="Username Hash"
format="Text" level="IndexValues" token="true"/>
<key name="ip.src" description="Source IP Address" format="IPv4"
level="IndexValues" protected="true"/>
<key name="ip.src.hash" description="Source IP Address Hash"
format="Text" level="IndexValues" token="true"/>
</language>
```

d. To restart the Concentrator service, go to **(Admin)** > **Services**, find the concentrator service you want to restart, and select > **Restart**.
The service should automatically restart.

Configure Data Retention on Concentrators and Decoders

Data retention configuration ensures that the data residing in the NetWitness Core components is deleted after a certain time. Configuring data retention on Concentrators and Decoders is not required for all environments, but it may be necessary to be in compliance with applicable laws and regulations. It is important to evaluate an appropriate retention period for your environment. The Data Retention Scheduler settings that you set apply to ALL data on a Concentrator or Decoder.

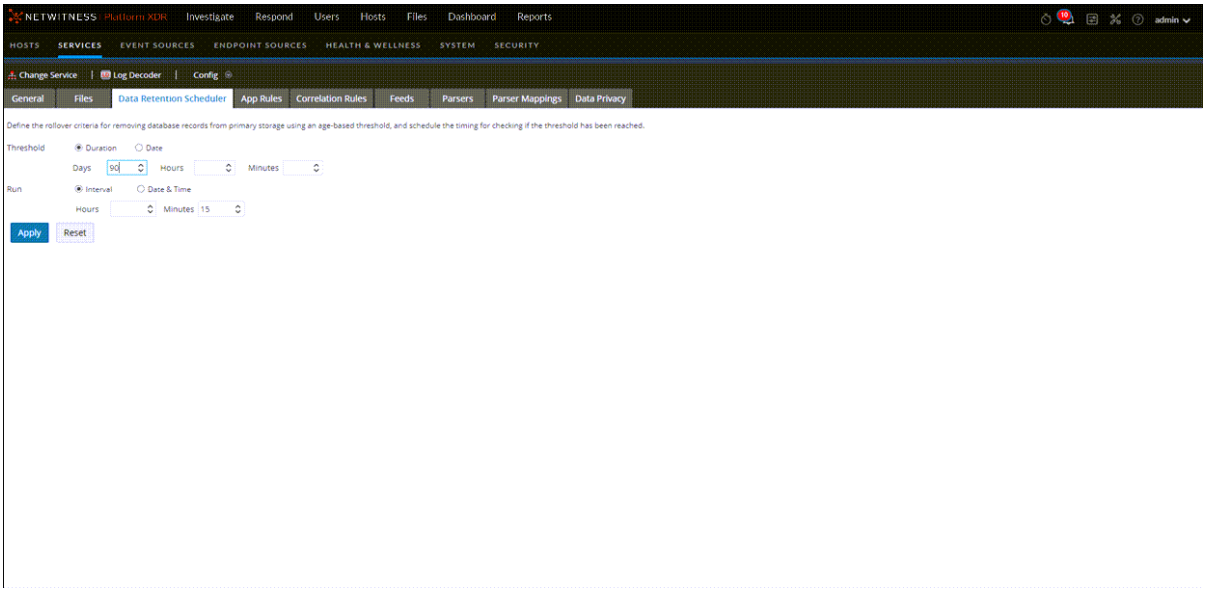
In the following example, NetWitness is configured to execute a check every 15 minutes to determine if the duration threshold has been met. If the threshold is met, NetWitness deletes data older than 90 days in the relevant databases.

Caution: The 90 day retention period is just an example. Adjust your rollover criteria depending on the location of the data and the applicable laws. In a strict data privacy environment, such as in Europe where laws require that Personally Identifiable Information (PII) not be saved or removed frequently, you may need to adjust the time.

This procedure is optional. If you do not set a time retention limit, the system automatically deletes the oldest data when the hard drive space is full.

(Optional) For each Concentrator and Decoder:

1. Navigate to the **Services Config** view **> Data Retention Scheduler** tab.



2. Define the data retention period. For example, set the **Threshold** to **Duration**, and type **90** in the **Days** field.
3. Define how often the scheduler checks to see if the threshold has been met. For example, set the runtime to **Interval** and select **15** in the **Minutes** field.
4. To save the configuration, click **Apply**.

Validate Data Privacy Protection

At this point, users have been added with roles that have permissions around specific types of metadata. The next step is to make sure the restricted user (the analyst) cannot view what the unrestricted user (the DPO) can. Also you need to ensure that the data retention configuration is limiting how long data is kept on the systems.

1. View role-based obfuscation in action:
   a. Log on as the unrestricted user (DPOadmin) and make sure this user can see all the data including the protected sensitive data `username` and `ip.src` along with any session that contains that metadata.
   b. Log off and the back on as the DPO user.

c. For each Decoder and Log Decoder, import a PCAP or logfile into the Services System view. Go to  **(Admin)** > Services, click a service, and use the **Upload Packet Capture File** option to upload a PCAP file that contains `username` and `ip.src` metadata.

d. When the import is complete, go to **Investigate > Navigate**, select the Concentrator connected to the Decoder to which the data was just imported.

e. Scroll down to make sure the `username` and `ip.src` meta keys and corresponding values are visible.

f. Click one of the green numbers next to a `username` or `ip.src` value and verify that the session loads in the Events view.

g. Make a note of the session ID to check when logging on as the restricted user.

h. Log off and log on as the restricted user (NonprivAnalyst).

i. Repeat steps c through f to verify that the user cannot see any `username` or `ip.src` metadata or sessions with that metadata including the one previously mentioned.

j. To jump to a specific session in the **Navigate** view, in the **Actions** menu, select **Go to Event** and enter the session ID.

2. Validate that the data retained in the database falls within the retention time configured in the Data Retention Scheduler.

a. Log off and log on as the unrestricted user (DPOadmin).

b. On the Concentrator, navigate to the **Services > Explore** view.

c. In the node tree, select the **database** node and then **stats**.

d. Observe the `meta.oldest.file.time` value and verify that this is not older then the threshold put on the data retention scheduler.

e. Change the service to the Decoder and repeat steps b through d, check for `stats meta.oldest.file.time` and `packet.oldest.file.time`.

In-Depth Procedures

This topic is a collection of procedures that a Data Privacy Officer uses to implement a data privacy plan for the NetWitness network. These procedures are part of an overall configuration, and are performed as needed to implement the data privacy plan and manage the flow of information in the network.

- Configure Data Obfuscation
- Configure Data Retention
- Configure User Accounts for Use in Data Privacy

Configure Data Obfuscation

This topic provides the procedures for configuring data obfuscation in NetWitness. In a single deployment, all Core service configurations for a data privacy solution must be the same; be sure to use the same hash and salt across all Decoders and Log Decoders.

Note: In order for data obfuscation to work, user accounts need to be configured as described in Configure User Accounts for Use in Data Privacy.

Configure the Decoder Hash Algorithm and Salt

Value hashing accomplished as part of the data privacy solution occurs at the time of meta key creation on the Decoder and Log Decoder. Both services have default settings for use with all meta keys whose values are transformed without a specified hash algorithm type or salt value. The initial NetWitness values for defaults are: hash algorithm (SHA-256) and salt (none).

Note: The SHA-1 algorithm and it is not available.

If you want to change the default settings, you can edit them in the Services Config view > Data Privacy tab or in the following nodes in the NetWitness Services Explorer view:

- `/decoder/parsers/transforms/default.type`

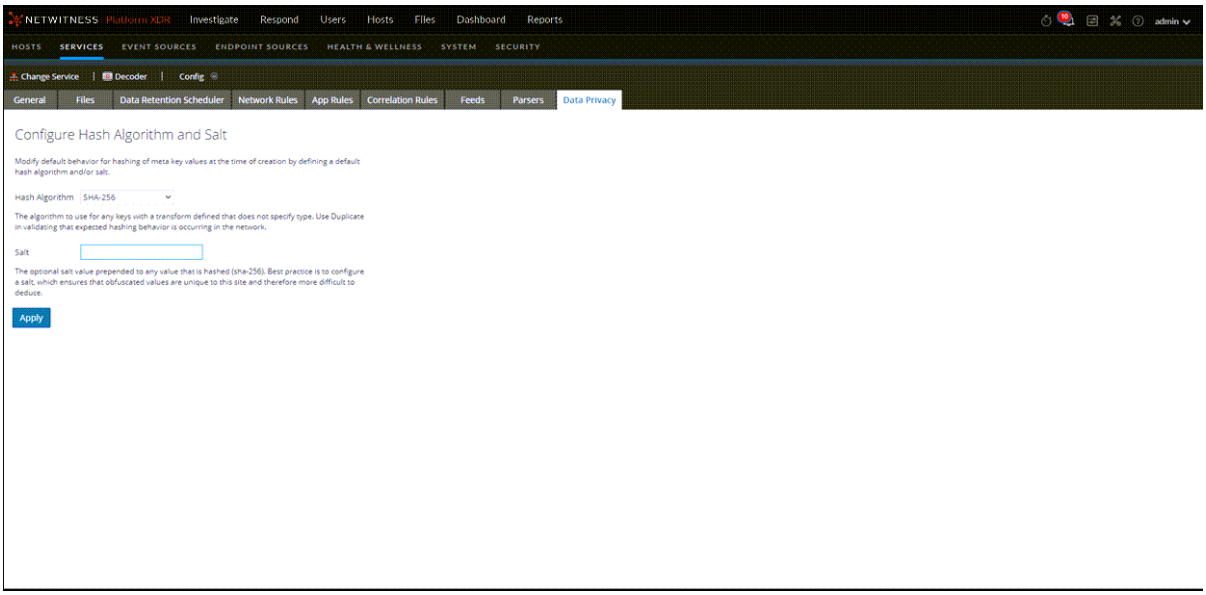
  The algorithm to use for any keys with a transform defined that does not specify `type`. The supported algorithms are: `duplicate` and `sha-256`.

- `/decoder/parsers/transforms/default.salt`

  The salt value prepended to any value that is hashed (`sha-256`). This value is optional, an empty salt is valid and produces an unsalted hash. The salt is not defined by default so that you can create a unique salt for your environment. In general, the longer and more complex the salt, the better the security. A salt of up to 60 characters can be used without any major impact. A salt of at least 16 characters is recommended.

To edit the default hash algorithm and salt

1. Go to  **(Admin) > Services**.
2. In the **Services** view, select a Decoder or Log Decoder service, and select  **> View > Config > Data Privacy**.



3. In the **Configure Hash Algorithm and Salt** section, select a **Hash Algorithm** to use for any meta keys with a defined transform that does not specify type: `sha-256`. (A second algorithm, `duplicate`, is available for administrators to use in validating that expected hashing behavior is occurring in the network.)
4. (Optional) In the **Salt** field, enter a salt value to be prepended to any value that is hashed. This value is optional, an empty salt is valid and produces an unsalted hash. The salt is not defined by default so that you can create a unique salt for your environment. In general, the longer and more complex the salt, the better the security. Best practices for security purposes dictate a salt value that is no less than 100 bits or 16 characters in length. If a unique salt is required for each individual meta key, that needs to be configured in the index file as shown in example 3 below.
5. Click **Apply**.

   The new settings become effective immediately.

Configure Language Keys

The NetWitness Core Language has several language key attributes to facilitate data privacy. You can edit these attributes in the custom index file for each Decoder or Log Decoder. The custom index file (for example, `index-decoder-custom.xml`) is editable in the Services Configuration view > Files tab. After making changes in the index file, like the ones shown in the examples below, a service restart in a specific sequence is required.

Based on the data privacy requirements for your site, configure individual meta keys to be protected using the following `key` attributes:

- `protected`

  This attribute specifies that NetWitness should consider the values as protected and tightly control any release of the value. When propagating the protected attribute, NetWitness ensures that any downstream trusted system treats the values accordingly. Add this attribute to all services that create the protected values (that is, Decoder or Log Decoder) and any services that will provide trusted access (software development kit (SDK) query/values, aggregation) outside of Core services. The exception to this rule is that a Broker with no index file specified does not need to have the attribute added.

- `token`

  This attribute specifies that values for this key are stand-ins for another value and may not be visually interesting. The `token` attribute is informational, primarily for UI elements to display the value in a more useful or visually pleasing format.

- `transform`

  This child element of `key` indicates that any values for a given meta key should be transformed and the resulting value persisted in another meta key. The `transform` element is only required on Decoders and Log Decoders and is informational if specified on any other Core services. The `transform` element has the following attributes and children:

Name	Type	Description	Optional or Required
`destination`	attribute	Specifies the key name where the transformed value will be persisted.	required
`type`	attribute	The transform algorithm to apply. If not specified, the value of `/decoder/parsers/transforms/default.type` is used.	optional
`param`	child-element	A name/value pair, where each `param` element has a required attribute `name` and the child text is the value. The only supported `param` is used to specify a meta key-specific `salt` value. If not specified, the following value is used: `/decoder/parsers/transforms/default.salt`	optional

Example 1

On a Decoder or Log Decoder, mark `username` as protected and hash all values into `username.hash` with the default algorithm and salt:

```
<key name="username" description="Username" format="Text" protected="true">
<transform destination="username.hash"/></key>
```

Example 2

On a Concentrator, mark `username` as protected and `username.hash` as token:

```
<?xml version="1.0" encoding="utf-8"?> <language level="IndexNone"
defaultAction="Auto">
<key description="Username" format="Text" level="IndexValues" name="username"
protected="true"/>
<key description="Username Hash" format="Binary" level="IndexValues"
name="username.hash"
token="true"/> </language>
```

Example 3

On a Decoder or Log Decoder, mark `username` as protected and hash all values into `username.bin` with the specified algorithm and salt:

```
<key name="username" description="Username" format="Text" protected="true">
<transform destination="username.bin" type="sha-256">
param name="salt">0000</param> </transform></key>
```

Configure Metadata and Content Visibility Per User Role on Core Services

On individual Broker, Concentrator, Decoder, Log Decoder, and Archiver services being viewed in the Services Security view (**Admin) > Services > Security**), administrators can configure the visibility of metadata and content based on the user group or role assigned to a user. This is called the SDK meta roles capability, and it is enabled by default. Administrators who want to configure metadata and content visibility per user must not disable the `sdk.content` permission (in the Roles tab). If the `sdk.content` permission has been disabled in the Roles tab, packets and raw logs are not visible to `system.roles` node. The `system.roles` node handles the filtering using the method configured in the Settings tab.

With `sdk.content` capability enabled, the next step is to select the method of filtering metadata and content in the Settings tab. Selecting a blacklist or whitelist option makes additional permissions for specific meta keys available in the Roles tab. The result is that administrators can choose a user role, such as analyst, in the Roles tab and select specific meta keys (and content) to be blacklisted or whitelisted for that user group. The permissions apply to any user in the user group.

For the Events view to display events correctly, the `sdk.meta.medium` meta key must be enabled to display values in the Type column (one of the default columns in the Events list). In addition, any meta keys used in a Query Prefix under User Settings in the Users tab or a PreQuery configured for Investigate must be enabled (**Admin) > Security > Users** or **(Admin) > Security > Roles**). If the meta key is not enabled for the user group, the column will not be displayed in the Events list.

Note: In the Version 11.4 and later Events panel, events are listed even when the user does not have permission to view the event reconstruction. In addition, the download options may still be available when a user does not have permission to view the event reconstruction. In these cases the user is allowed to download the events, but the output is an empty file because restricted permissions are enforced during the download.

The following table lists the options for filtering in the Settings tab and the numeric values used to disable (0) and the types of filtering (1 through 6). There is no need to know the numeric value unless configuring metadata and content visibility manually in the `system.roles` node.

system.roles Node Value	Settings Tab Option	Event Metadata	Original Event
0	**No Filtering.** System roles that define permissions on a per meta key basis are disabled.	Visible	Visible
1	**Whitelist meta and content.** By default no meta keys and no packets are visible. Selecting individual SDK meta roles per user group allows users to see metadata and packets for that SDK meta role.	Not Visible-Select to Show	Not Visible-Select to Show
2	**Whitelist only meta.** By default packets are shown, but no metadata is visible. Selecting individual SDK meta roles per user group allows users to see metadata for that SDK meta role.	Not Visible-Select to Show	Visible
3	**Whitelist only content.** By default metadata is visible, but no packets are visible. Selecting individual SDK meta roles per user group allows users to see packets for that SDK meta role.	Visible	Not Visible-Select to Show
4	**Blacklist meta and content.** By default all metadata and all packets are visible. Selecting individual SDK meta roles per user group prevents users from seeing metadata and packets for that SDK meta role.	Visible-Select to Hide	Visible-Select to Hide
5	**Blacklist only meta.** By default all metadata and all packets are visible. Selecting individual SDK meta roles per user group prevents users from seeing metadata for that SDK meta role.	Visible-Select to Hide	Visible
6	**Blacklist only content.** By default all metadata and all packets are visible. Selecting individual SDK meta roles per user group prevents users from seeing packets for that SDK meta role.	Visible	Visible-Select to Hide

Three factors determine what a user sees:

- The SDK meta role setting (blacklist or whitelist).
- The restricted meta keys configured for the group to which the user belongs.

- The meta keys in the session being analyzed. These include meta keys specified in a query prefix for a user at the service level, and for roles and users at the system level.

Caution: Be aware that with blacklisting, implicit trust is granted for all except the configured metadata. For a Decoder to have RBAC enabled and use implicit trust, it must only use a blacklist system setting; a whitelist setting will result in some issues with meta keys that are not explicitly enabled and therefore not visible. It is impossible to grant implicit trust under whitelist rules because the universe of meta keys cannot be known. If you want to use whitelisting, a workaround is to turn RBAC off for the Decoder and disable any user accounts from connecting directly to the Decoder if they should use RBAC.

Here is an example of how the SDK meta role configuration meshes with a Group that has restricted meta keys.

Configuration:

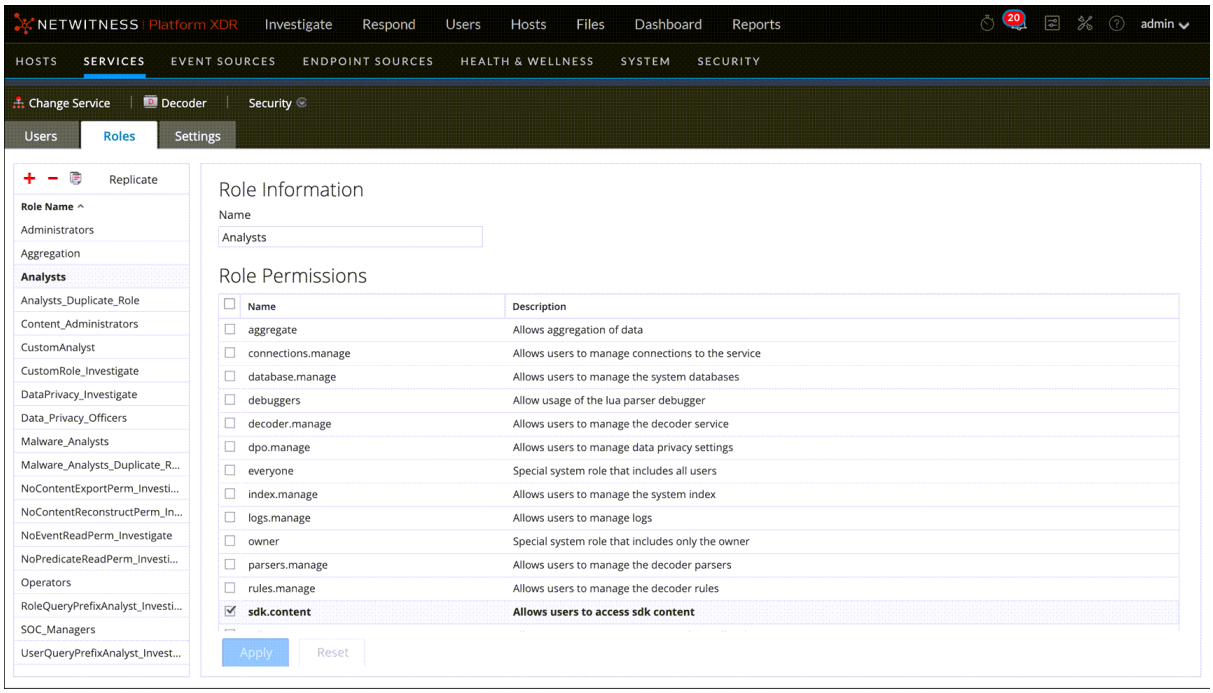
- The SDK meta role setting is **Blacklist meta and content**. With this option implemented, all metadata and all content (packets and logs) are visible by default.
- The administrator has restricted meta keys configured for the Analysts group to prevent viewing of sensitive data (for example, `username`).
- The packets and logs for any session that includes the `username` meta key are not visible to an analyst.

Result: Now a user who is a member of the Analyst Group investigates a session. Depending on the content of the session, the results are different:

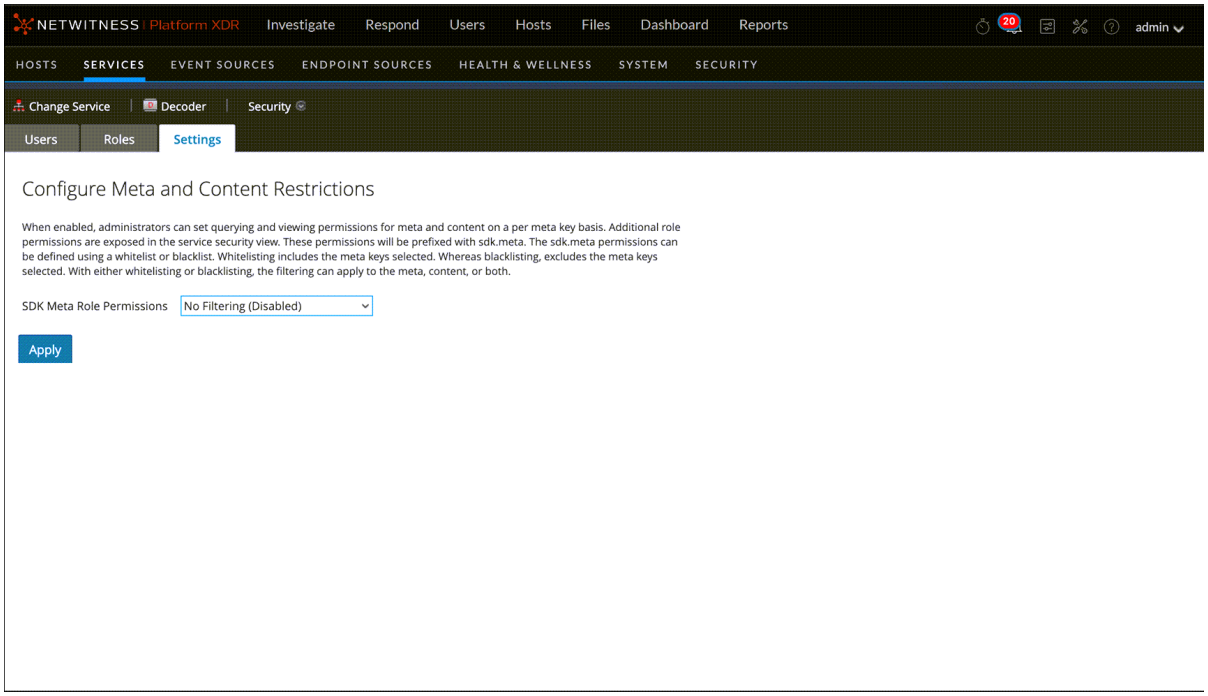
- Session 1 includes the following meta keys: `ip`, `eth`, `host`, and `file` The session does not include `username` so all packets and logs in the session are displayed.
- Session 2 includes the following meta keys, `ip`, `time`, `size`, `file`, and `username`, Because the session includes `username` , no packets or logs from the session are displayed for the analyst.

To configure metadata and content restrictions for a Decoder or Log Decoder:

1. Go to  **(Admin)** > **Services**.
2. In the **Services** view, select a Broker, Concentrator, Decoder, Log Decoder, or Archiver service and select  **> View > Security**. Click the **Roles** tab, select a role, and verify that the `sdk.content` permission is enabled.

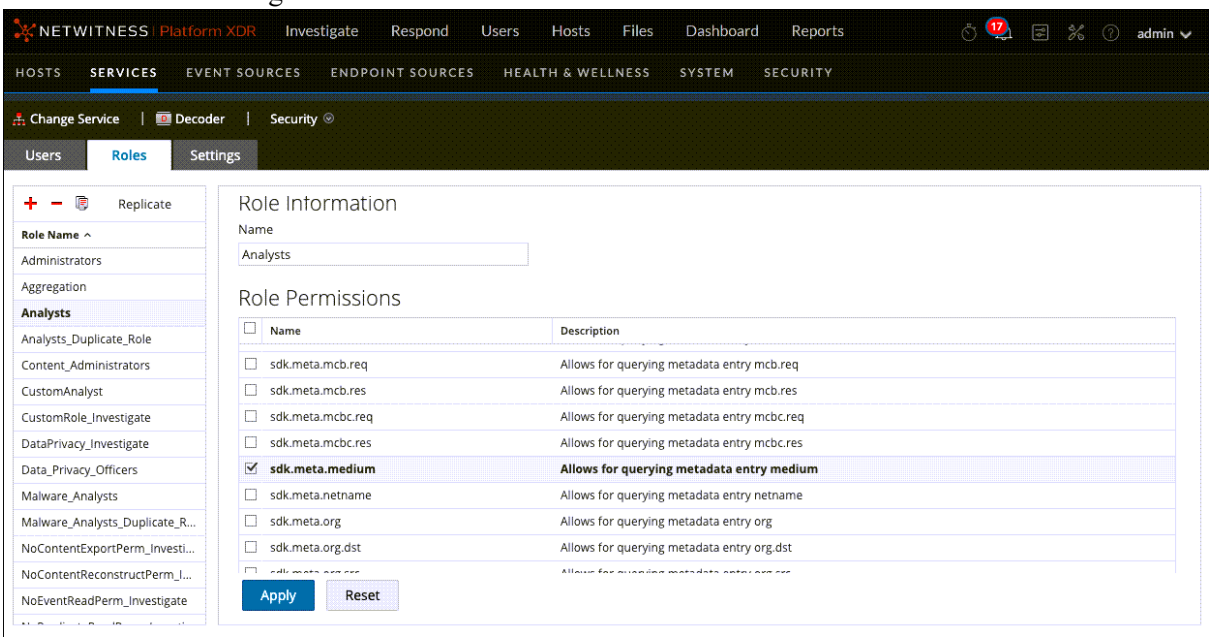


3. Click the **Settings** tab.



4. Select one of the filtering methods (blacklist or whitelist) and content types (meta and content, meta only, or content only), and click **Apply**.
5. Click the **Roles** tab and a role for which you want to allow content (whitelist) or block content (blacklist) as specified in the SDK Meta Role Permissions setting.

The Role Permissions for the selected role are displayed, and the SDK Meta Role Permissions are available for selection, for example, `sdk.meta.medium`. If you selected one of the whitelist options in the SDK Role Permissions setting, you must assign each SDK meta role to make the selected content visible to users assigned that SDK meta role. The `sdk.meta.medium` permission must be enabled in order for the Type column to be displayed in the Investigate > Events view. If you selected one of the blacklist options in the SDK Role Permissions setting, selected content will be hidden from users assigned that SDK meta role.



6. Select the SDK meta role permissions for users assigned this role. Click **Apply**.

The settings become effective immediately and apply to new packets and logs processed by the Decoder or Log Decoder.

Configure Meta Keys Not Written to Disk Per Parser on a Decoder

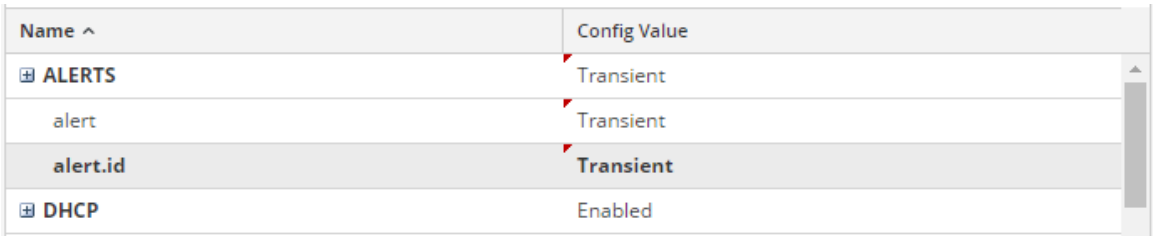
On a Decoder and Log Decoder, a Data Privacy Officer can configure individual meta keys that are not written to disk. To do so, the DPO specifies the meta keys as transient in the index and the parser configuration.

Note: The same capability was previously available on Log Decoders, and was configured when setting up parsers by modifying the table-map.xml file. Now it is integrated in the Services Config view.

To configure selected meta keys on individual parsers that will not written to disk:

1. Go to **(Admin)** > **Services**.
2. In the **Services** view, select a Decoder or Log Decoder service and select **> View > Config**.
3. In the **Parsers Configuration** section of the **General** tab, select a parser and then select **Transient** in the **Config Value** drop-down list. Access the list by clicking on the configuration value (Enabled, Disabled, or Transient).

The configuration change is marked by a red triangle.



4. Click **Apply**.

   The change is effective immediately. The parser configured as Transient will no longer store meta keys to disk.

Configure Data Retention

A NetWitness user with the role of Administrator can configure NetWitness to ensure that sensitive data has been removed after a specific retention period, regardless of system ingest rate. For instance, the policy might be to keep packets (both raw data and metadata) for no more than 24 hours, and to keep some logs (both raw data and metadata) for up to seven days. If sensitive data makes its way into another database on the Reporting Engine, Malware Analysis, Event Stream Analysis, and NetWitness Servers, data retention can be managed there as well. The administrator needs to set up each service individually across all NetWitness components (except Event Stream Analysis) based on policy and data privacy laws.

Sensitive data may also be in cache.

- Brokers can cache data and this needs to be cleared by configuring an independent rollover and other removal of cache as required. The administrator can configure cache rollover for a Broker by editing the `Scheduler` file in the Services Config view Files tab.
- Investigate and the NetWitness Server cache data, and this is cleared automatically every 24 hours.
- If the Data Privacy Officer (DPO) exports data, that is the same as saving data on the NetWitness Server in the jobs queue. To clear this data, the administrator or DPO should clean up the jobs queue on a regular basis.

Data Retention

You can schedule a recurring job for Decoder, Log Decoder, and Concentrator services in NetWitness to check if data is ready to be removed. The Data Retention Scheduler provides a means to configure basic scheduling (see below), and advanced Scheduler settings are also available by editing the `Scheduler` file in the Services Config view Files tab or the node in the Explorer view.

The Archiver has flexible data storage and retention options. You can place different types of log data into individual collections and manage them separately. These collections enable you to specify how much of the total storage space to use and how many days to store the logs in the collection. You can also determine whether to delete the log data or to move it to offline cold storage after it reaches the maximum specified storage space for the collection.

For example, you can put sensitive information in a collection and configure a limitation on how long to keep it, such as 30 days. To delete the data after 30 days, you would not enable warm or cold storage for that collection.

Deleting versus Retaining Log Data

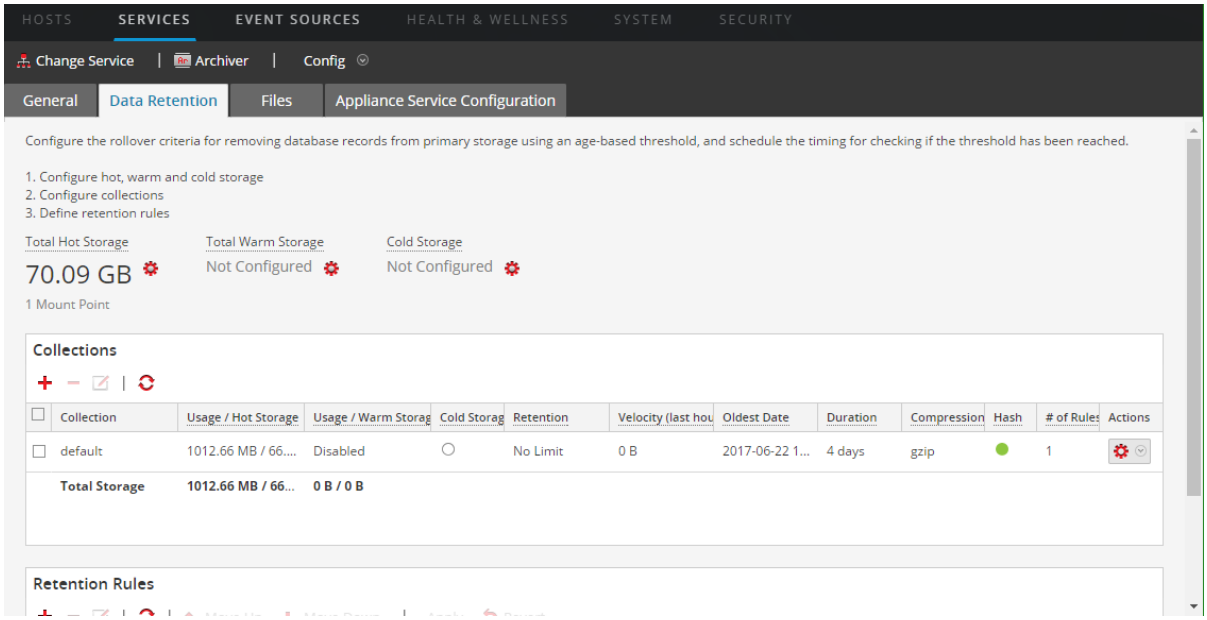
Administrators can configure hot, warm, and cold tiered storage on an Archiver. Cold storage contains the oldest log data that is either required for the operation of the business or mandated by regulatory requirements. When a collection reaches its retention limits for hot and warm storage, NetWitness deletes the log data from hot or warm storage. With cold storage configured, a copy goes into cold storage before the logs are deleted from hot or warm storage. You can choose whether to enable cold storage for each log storage collection. NetWitness does not manage cold storage.

Enable or Disable Cold Storage in a Log Storage Collection

When log data in a collection reaches its retention limits for hot and warm storage, you can delete it or move it to offline (cold) storage.

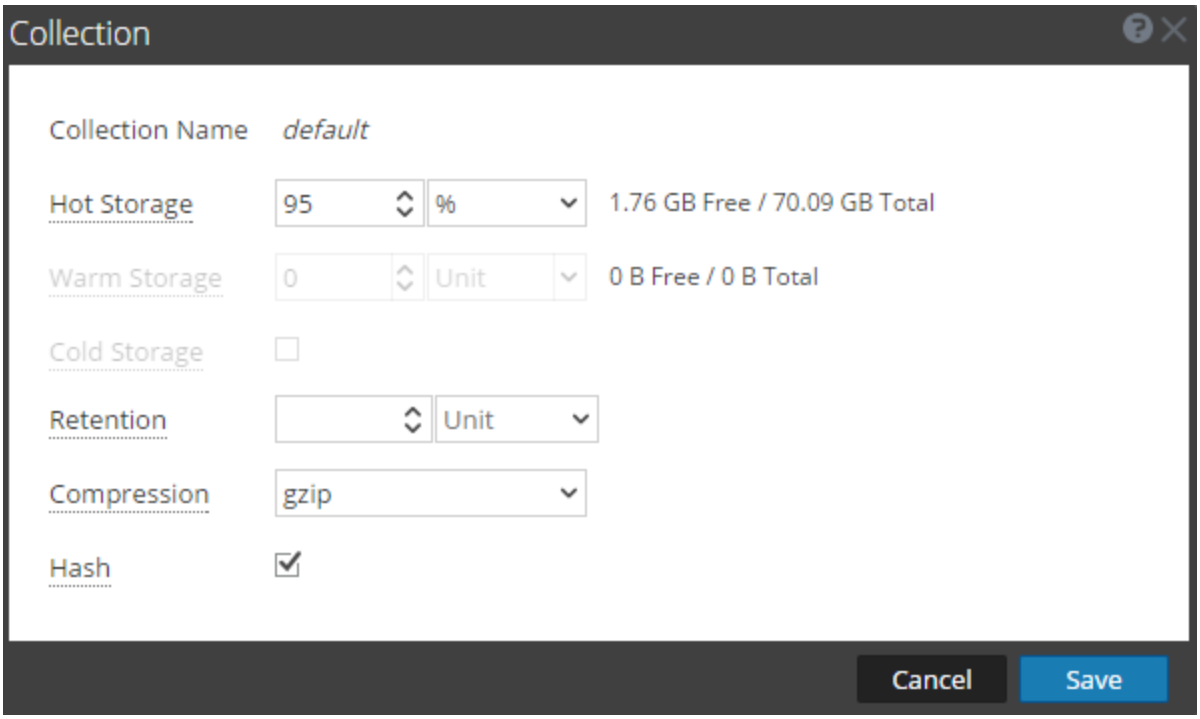
To enable or disable cold storage in a log retention storage collection on an Archiver:

1. Go to **(Admin)** > **Services**.
2. Select the Archiver service and select **> View > Config**.
3. Click the **Data Retention** tab.



4. In the **Collections** section of the Data Retention tab, select a collection and click .

   The Collection dialog is displayed.



Note: If the maximum storage size of the collection does not allow full data retention for the retention period specified, NetWitness deletes the data or it goes to warm or cold storage if specified in the collection.

5. Enable or disable cold storage:
   - To delete log data when the collection reaches its specified retention limits, clear the **Cold Storage** checkbox.
   - To move log data to offline storage when the collection reaches its specified retention limits, select the **Cold Storage** checkbox.
6. Click **Save**.

Configure Log Retention and Storage on an Archiver

To configure log retention and storage on an Archiver, see "Configure Archiver Storage and Log Retention" in the *Archiver Configuration Guide*.

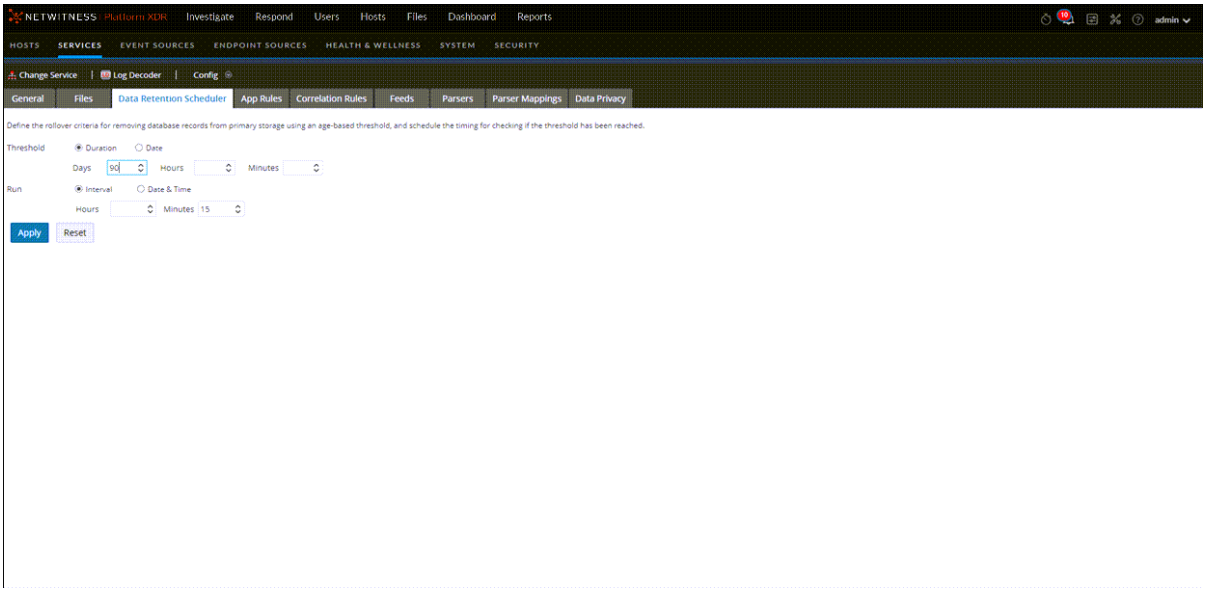
Schedule a Recurring Job to Check Data Retention Thresholds

The data retention scheduler configuration ensures that the data residing in the Decoder, Log Decoder, and Concentrator components is deleted after a certain time. For example, data retention on a Decoder might be configured to execute a check every 15 minutes to determine if the specified duration threshold has been met. If the threshold is met, NetWitness deletes data older then 4 hours in the relevant databases.

Caution: The schedule overwrites any previous schedule and becomes effective immediately. If the retention period is decreased, the data exceeding this retention period is removed.

For a Decoder, Log Decoder, or Concentrator:

1. Go to **(Admin)** > **Services**.
2. In the **Services** view, select a Decoder, Log Decoder, or Concentrator service and select **> View > Config**.
3. Click the **Data Retention Scheduler** tab.



4. Set the threshold based on the duration of time the data has been stored or the date on which the data was stored. Do one of the following:
   a. To define the duration of time that data can be stored before removal, select **Duration**, and then specify the number of days (365 maximum), hours (24 maximum), and minutes (60 maximum) that have elapsed since the time stamp on the data.
   b. To define the removal of data based on the date of the timestamp, select **Date**, and then specify the monthly date and time in the Calendar and Time fields.
5. Do one of the following to configure the **schedule for checking rollover criteria**:
   a. If you want to set a regular interval at which the scheduled database check occurs, select **Interval** and specify the **Hours** and **Minutes** between the scheduled checks.
   b. If you want to set a regular date and time at which the scheduled database check occurs, select **Date and Time** and specify the system clock time in `hh:mm:ss` format for the rollover.
      - To specify the day, select **Every Day**, **Weekdays,** or **Weekends**. The Scheduler defaults to **Every Day**.
      - To specify a different set of days of the week, select **Custom** and click on each day on which the database check occurs.

      Caution: The schedule overwrites any previous schedule and becomes effective immediately. If the retention period is decreased, the data exceeding this retention period

is removed.

6. Click **Apply** to complete the configuration.

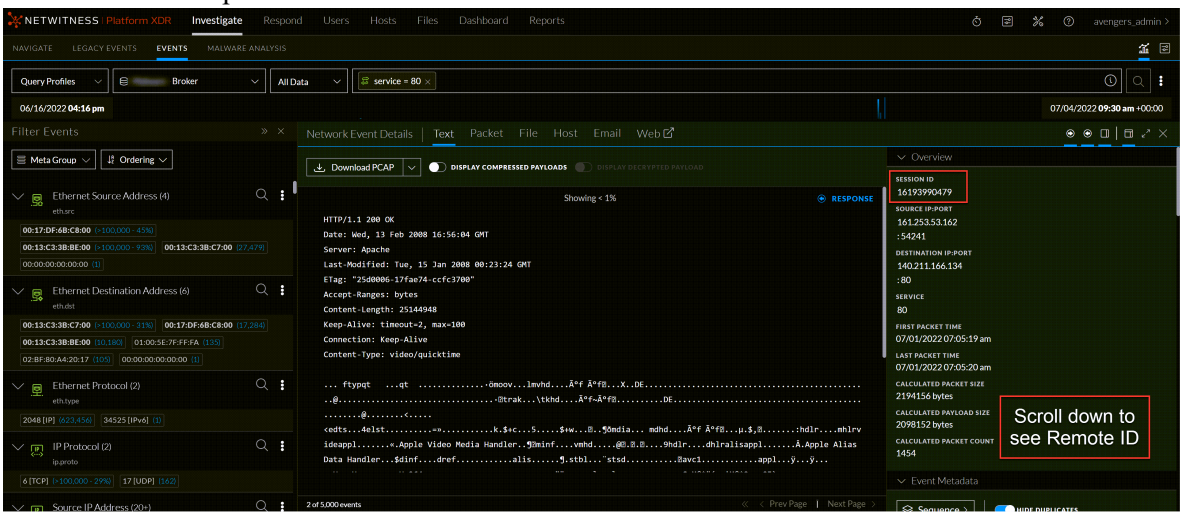
Purge Data Using String and Pattern Redaction Option

If you have data where you do not want it to be (for example, you have classified data in your unclassified Network Decoder), purging data using string and pattern redaction is the fastest way to wipe data. It is ideal for smaller amounts of data. This method requires that you identify the data to be purged by using a query in either the Investigate view, the REST API interface, or the NwConsole. Using the Investigate view requires additional steps to clear user interface cache on the admin server. This procedure includes the extra steps needed when you find the data to be purged using the Investigate view.

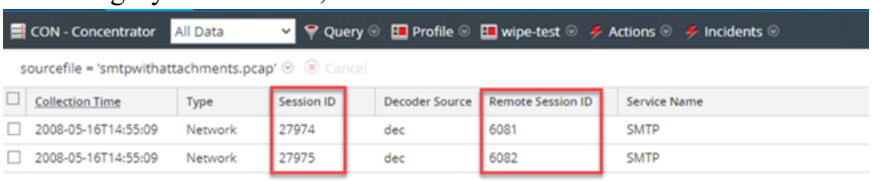
Note: If a large amount of data needs to be purged, it is best to start with the data on the storage component. Refer to Optional Data Overwriting Options for additional information.

To purge the data:

1. Go to **Investigate** > **Events** or **Investigate** > **Legacy Events** and find the events that you want to purge. In the Events view you can see the session ID in the Overview panel. You need to scroll down in the Event Meta panel to see the remote ID.

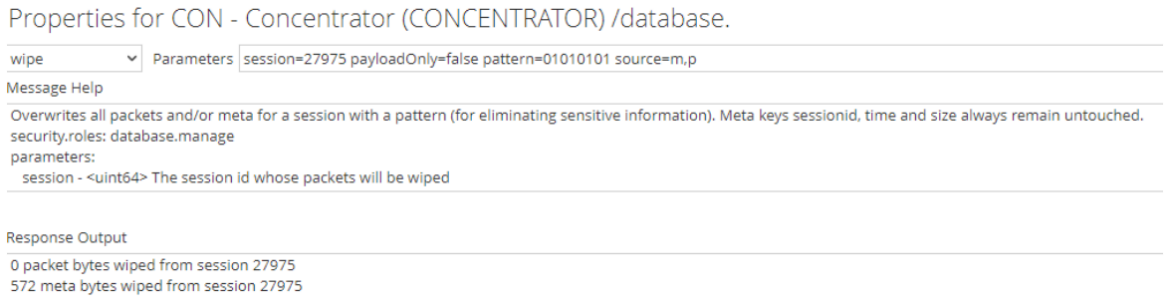


In the Legacy Events view, both IDs are visible in the Events list.

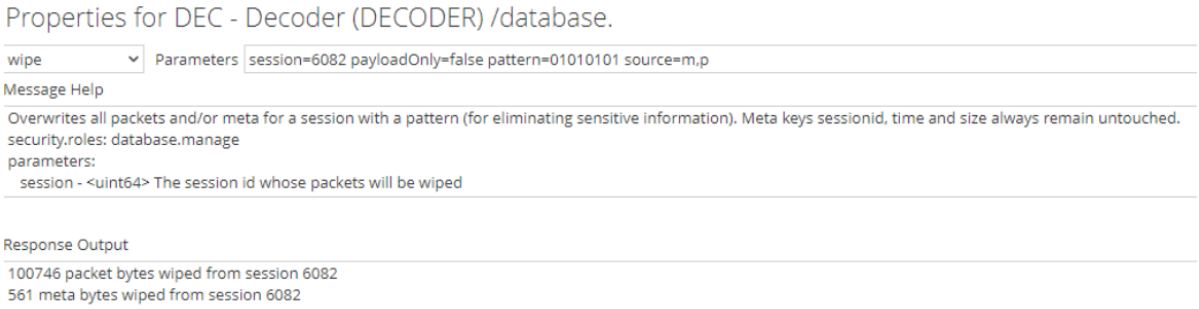


2. Make a note of the session ID and the remote ID of each event to be deleted. In the example above, the session IDs are 27974 and 27975. The remote IDs are 6081 and 6082.

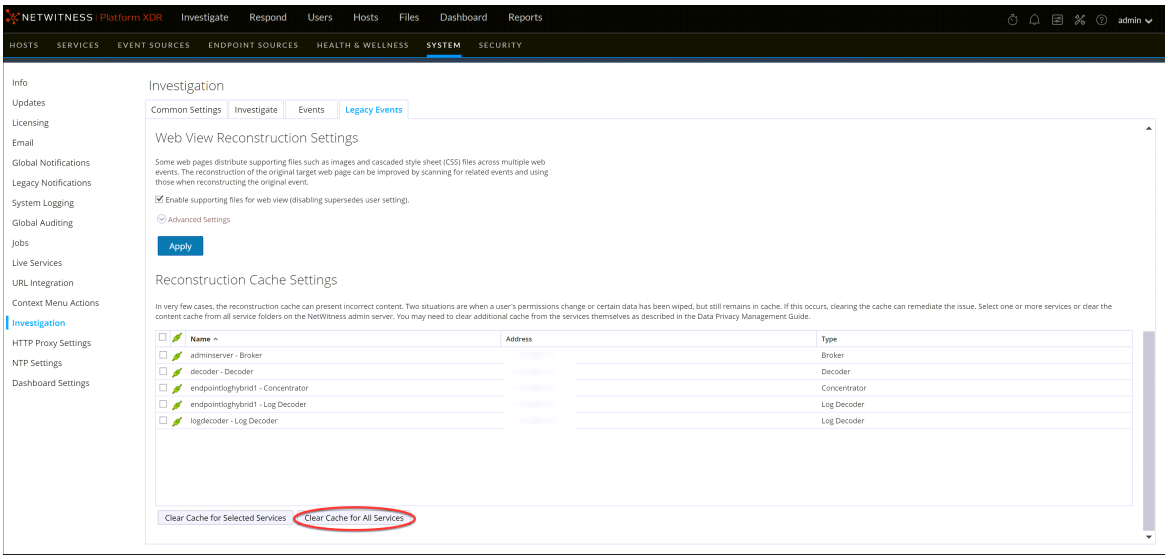
3. Starting with the Concentrator, view the service using the REST API. Refer to the *RESTful API User Guide* for instructions. Go to the NetWitness All Versions Documents page and find NetWitness Platform guides to troubleshoot issues.
   In the REST API, enter the `wipe` command with the following parameters:
   - **`session=<uint64>`**: The session ID whose packets will be wiped.
   - **`payloadOnly=<bool, optional>`**: If `payloadOnly` is `true`, only the packet payload will be overwritten. The default value is `true`.
   - **`pattern=<string, optional>`**: The pattern uses all zeros by default. If you use a string as your pattern, it will not overwrite any meta values that are not a string type. Therefore, it is best to keep the pattern as a numerical value.
   - **`metaList=<string, optional>`**: Comma-separated list of metadata to wipe. The default (empty) is all metadata.
   - **`source=<string, optional, {enum-any:m|p}>`**: The types of data to wipe: metadata, packets, or both. The default is packets only. The best option here is to wipe both metadata and packets `(m,p)`.



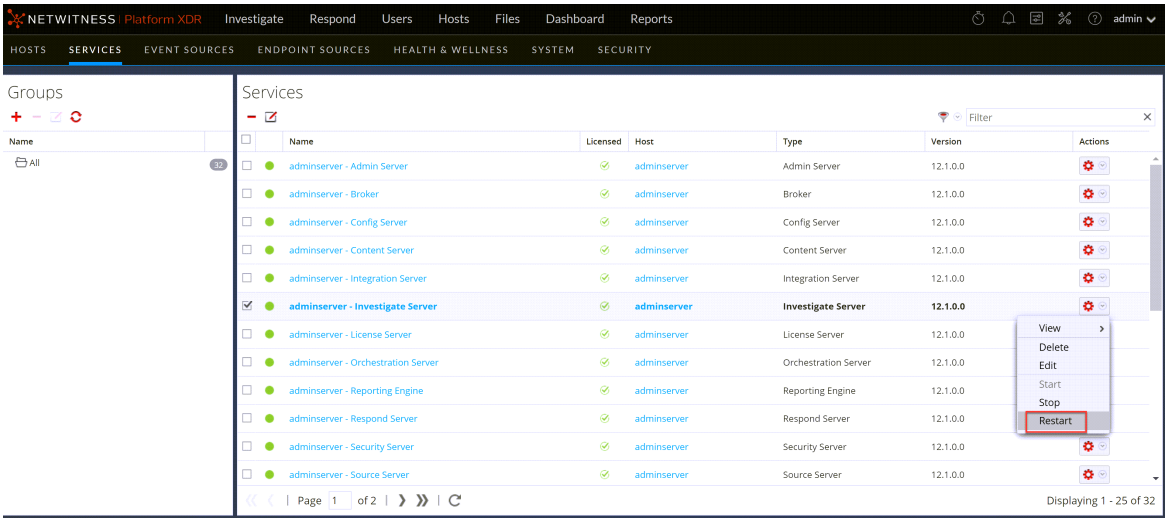
4. For each Network Decoder and Log Decoder in the path of the query, use the `wipe` command and the remote session IDs to overwrite the raw sessions on disk.



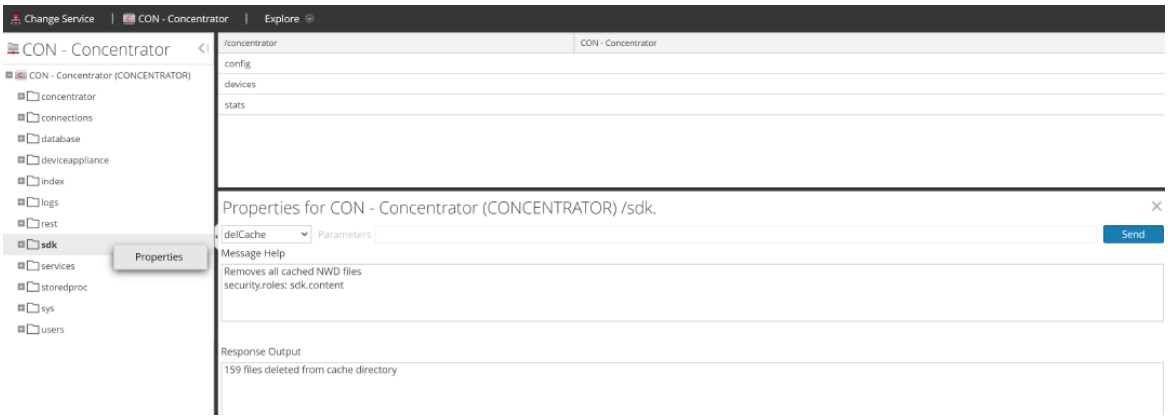
5. To ensure that the indexed meta values that were stored on the Concentrator index are removed, rebuild the index. This can take a long time but is necessary because the `wipe` command does not remove any data from the Concentrator index. Refer to the *Core Database Tuning Guide* for instructions.
6. (Optional) If the data was discovered using the Investigate view, additional steps are necessary:
   a. To clear the query path to prevent an analyst from bringing up an old copy of the session, go to: **(Admin)** > **System** > **Investigation** > **Legacy Events** and click **Clear Cache for All Services**.



b. To clear the cache for a reconstruction in the Events view, restart the Investigate service.



c. To clear the cache for the Concentrator and Network Decoder, go to  **(Admin)** > **Services** > **Concentrator** > right-click **sdk** > **Properties** and  **(Admin)** > **Services** > **Decoder** > right-click **sdk** > **Properties** and execute the `delCache` command. The figure below shows this path for the Concentrator.



An analyst attempting to view the same session in Investigate will see that the raw data is unavailable for viewing and the metadata has been overwritten.

Configure User Accounts for Use in Data Privacy

This topic provides the procedures for configuring user accounts that work with data obfuscation in NetWitness. In order for data obfuscation to work, accounts and permissions for several types of users must be configured.

- Customize the default Administrators system role in NetWitness to remove permissions that should be available only to the Data Privacy Officer.
- Add two new user accounts at the system level to depict a data privacy officer and a typical analyst.
- Add a user account at the service level with the aggregation role so that Decoders and Log Decoders can aggregate data to a Concentrator or Broker.
- On the Reporting Engine, configure two separate service accounts. One service account for general purpose reporting that does not include any sensitive data and the other account for privileged users with access to all data including sensitive data. This procedure is described in "Configure Data Source Permissions" in the *Reporting Engine Configuration Guide*.

Customize the Default Administrators User Role at the Service Level

To separate the data privacy officer and administrator functions on each Decoder and Log Decoder, you need to remove the `dpo.manage` permission from a clone of the Administrators role.

1. Go to  **(Admin)** > **Services**, select a Decoder or Log Decoder, then select  **> View > Security**.
2. In the **Services Security** view, click the **Roles** tab, select **Administrators** and click  .
   In the **Enter Role Name** dialog, enter a new role name such as Non_DPO_Administrators and click **Save**.
3. Select the new role.

   The Role Information is displayed for editing.
4. Clear the box next to **dpo.manage** so that it is no longer checked and click **Apply**.

   The permission to manage data privacy configuration is removed for the new role.
5. In the **Users** tab, select each user who has the **Administrators** role, and change their role to the cloned role.
6. Validate that the users with the modified Administrators role can login as with admin privileges.
7. Validate that the users with the modified Administrators role cannot configure metadata and content restrictions in the Settings tab.

Add a User Account with the Aggregation User Role at the Service Level

To ensure that Decoders and Log Decoders can aggregate data to a Concentrator or Broker:

1. **Go to** **(Admin) > Services, select a Decoder or Log Decoder, then select** **> View > Security.**
2. In the **Users** tab, add a user with the `Aggregation` role and click **Apply**.

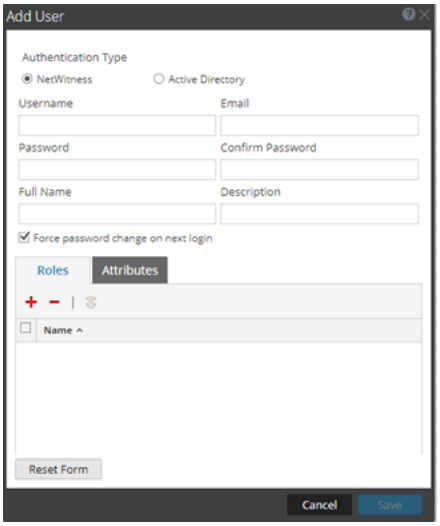
Note: "Aggregation Role" in the *Hosts and Services Getting Started Guide* provides details about the application of this user role.

Add Data Privacy Officer and Analyst Accounts on the NetWitness Server

You need to add two new user accounts in NetWitness at the system level to depict a privileged data privacy officer and a typical analyst. If the environment is configured using the default trusted connections, you do not need to create the new user accounts on the Core services (Brokers, Concentrators, and Decoders). When a user is created in the NetWitness Server, that user can log on to the services.

Note: The role name is required to exist on both the server and the services, and the role name for all places must be identical. If you create a new custom role on the NetWitness Server, make sure to add it to all Core services as well.

1. Create a new user account for the data privacy officer:
   a. **Go to** **(Admin) > Security,** select the **Users tab** and click **.**
   The Add User dialog is displayed.



   b. Create the new account with the following credentials.

   Username = <new user name for logon, for example, DPOadmin>
   Email = <new user's email, for example, DPOadmin@rsa.com>
   Password = <new user's password for logging on, for example, RSAprivacy!@>
   Full Name = <new user's full name, for example, DPO Administrator>

c. Click the **Roles** tab,  , and select the `Data_Privacy_Officers` role for the new user.

d. Select **Save**.

2. Create a new user account for the analyst with limited privileges:

a. In the  **(Admin) > Security** view, click the **Users** tab. In the **Users** tab toolbar, click  .

The Add User dialog is displayed.

b. Create the new account with the following credentials:

Username = <new user name for logon, for example, NonprivAnalyst>
Email = <new user's email, for example, NonprivAnalyst@rsa.com>
Password = <new user's password for logging on, for example, RSAprivacy!@>
Full Name = <new user's full name, for example, Nonprivileged Analyst>

c. Click the **Roles** tab,  , and select the `Analysts` role for the new user.

d. Select **Save**.

Data Privacy References

The following reference materials are available for management of data privacy and data retention. Go to the NetWitness All Versions Documents page and find NetWitness Platform guides to troubleshoot issues.

- See "Data Privacy Tab" in the *Decoder and Log Decoder Configuration Guide.*
- See "Data Retention Tab - Archiver" in the *Archiver Configuration Guide.*
- See "Data Retention Scheduler Tab"in the *Hosts and Services Getting Started Guide.*