One Identity Safeguard for Privileged Sessions 6.9.4

Safeguard for Privileged Analytics Configuration Guide

Legend

WARNING: A WARNING icon highlights a potential risk of bodily injury or property damage, for which industry-standard safety precautions are advised. This icon is often associated with electrical hazards related to hardware.

CAUTION: A CAUTION icon indicates potential damage to hardware or loss of data if instructions are not followed.

SPS Safeguard for Privileged Analytics Configuration Guide
Updated - 27 January 2022, 02:50
Version - 6.9.4

Contents

1

Introduction

This guide walks you through the steps required to configure One Identity Safeguard for Privileged Sessions (SPS) so that you can start analyzing session data and user behavior using One Identity Safeguard for Privileged Analytics (SPA).

SPS and SPA are part of the One Identity Safeguard solution, which in turn is part of One Identity's Privileged Access Management portfolio.

One Identity Safeguard for Privileged Sessions (SPS) integrates data from SPS to use as the basis of user behavior analysis. SPA uses machine learning algorithms to scrutinize behavioral characteristics (using data from SPS), and generates user behavior profiles for each individual privileged user. SPA compares actual user activity to user profiles in real time, with profiles being continually adjusted using machine learning. When SPA detects unusual activity, this is indicated on the user interface of SPS in the form of high scores and visualized insight.

NOTE: The primary audience of this guide is One Identity Pre-Sales and Support Engineers, as well as Engineers representing One Identity's Partners.

If you wish to configure SPS to interwork with SPA as an end user, contact our Support Team or Professional Services for assistance.

2

Before you start

Prerequisites

One Identity Safeguard for Privileged Sessions has the following requirements when using it with One Identity Safeguard for Privileged Analytics:

Table 1: One Identity Safeguard for Privileged Sessions prerequisites

Type	Requirement
SPS version	Any supported version from version 5 F4 onward, ideally the latest one.
License	A license that has One Identity Safeguard for Privileged Analytics (SPA) enabled. To find out if your license supports SPA, obtain a support bundle, and check license information in the configuration XML. For details on how to obtain a support bundle, see "Collecting logs and system information for error reporting" in the Administration Guide. Alternatively, if you are unsure whether you have licensing enabled, it is safe to assume that you do not. NOTE: If you are using SPS 5 F5 or later, you are able to run SPA without a license option for 2 months.
Access rights	A user account with admin access rights.
Session data from network traffic	Session data that: • contains real, unique usernames linked to users other than root/administrator or a shared account • has commands extracted • has keystrokes extracted • has window titles extracted For more details, see Prerequisites in Analyze data using One Identity Safeguard for Privileged Analytics on page 13.

NOTE: If you are upgrading to SPS version 5 F4 or later from an earlier version, wait for the session database upgrade to finish.

To track progress, check the system monitor. It displays a message telling you that the session database upgrade is in progress, and it also shows the percentage of completion.

You can also go to **Search**, where all data that has been through the upgrade process is available.

In the case of large databases, the upgrade can take hours or even days, but the system should remain completely usable during the process. The upgrade starts with the most recent sessions and goes backward in time.

Limitations

SPS used in combination with SPA currently has the following limitations:

- SPA requires at least 12GB RAM to operate. If you are interested in upgrading your appliance, contact our Support Team.
- SPA requires a lot of computation, which can put pressure on SPS:
  - The keystroke algorithm is much more resource-hungry than the other algorithms, therefore our recommendation is to start analyzing data using the algorithms that require less resources.
  - Before you start using SPA, make sure that at least half the capacity of SPS is available.
- SPA only analyzes audit trails and SPS metadata, it does not analyze log data.

3

Algorithms

One Identity Safeguard for Privileged Analytics analyzes user behavior with the help of algorithms, also called analytics.

The algorithms of One Identity Safeguard for Privileged Analytics are mathematical methods that can be used to analyze session data from multiple angles. Algorithms have to be trained using a history of session data. Based on this training, an algorithm can build a baseline of a particular user's behavior and score new sessions. Scores will indicate whether a particular user's behavior is normal or unusual, compared to the baseline. Algorithms also provide visualization to display insight about user behavior.

Currently, the following algorithms are supported:

- The *keystroke algorithm* is able to tell whether a user is really who they say they are based on their typing dynamics. SPA compiles a typing profile for each user based on how many seconds it typically takes for the user to press combinations of keys on their keyboard. The keystroke algorithm analyzes keyboard data coming from RDP or SSH sessions and compares it with the user's profile.
- SPA compiles a commands profile for the user based on the commands that they usually execute. The *command algorithm* determines the probability of the occurrence of certain commands within a session.
- The *login time algorithm* builds a profile based on the exact time in a day when a user logs in. Based on the user's profile, it can tell how unusual the time of login is, given the daily distribution of the user's login events in the past.
- The *host login algorithm* analyzes how similar two hosts are based on the users that log in to those hosts. When a user logs in to a host that they never or only very rarely log in to, that will not be considered an anomaly if that host is similar to other hosts that the user frequently uses.
- The *frequent item set (fis) algorithm* is similar to a "customers who bought these items also bought" type of algorithm used on e-commerce websites. It examines multiple attributes of sessions and attempts to find values that frequently appear together, forming a set. Using this information, the fis algorithm is able to discover patterns in user behavior, such as "this person only uses RDP in the middle of the night from this IP address".
- The *window title algorithm* analyzes window titles to uncover unusual user behavior, that is, it identifies users based on what window titles they usually have on their

screen. It is currently an experimental algorithm and is disabled by default.

> **CAUTION:**
>
> **This is an EXPERIMENTAL feature. It is documented, but the performance impact on production systems has not been determined yet. Therefore this feature is not yet covered by support. However, you are welcome to try it (preferably in non-production systems) and if you have any feedback, send it to feedback-sps@oneidentity.com.**

- The *mouse-movement-based user authentication algorithm* is able to tell whether a user is who they say they are based on their mouse movements.

  > **CAUTION:**
  >
  > **This is an EXPERIMENTAL feature. It is documented, but the performance impact on production systems has not been determined yet. Therefore this feature is not yet covered by support. However, you are welcome to try it (preferably in non-production systems) and if you have any feedback, send it to feedback-sps@oneidentity.com.**

- The *scripted session detection* determines whether activities in a session point towards being a scripted session. The following internal algorithms in the background assist in determining whether a session is scripted:
  - The *clockmaster algorithm* is able to detect unnaturally precise sessions that start repeatedly at certain peak minutes of an hour (for example, at 8:30, 10:30, 11:30, and so on). The algorithm flags such sessions as scripted sessions. The reason behind this is that the minutes in the timestamps of humans' activities in a longer time period supposedly have random uniform distribution or are very close to it.
  - The *gapminder algorithm* is able to detect scripted sessions based on the time gaps between the sessions that belong to a given account. When the time gaps between sessions have typical, repeating values, then that suggests unnatural periodic behavior. The gapminder algorithm does not build baselines. Instead, it continuously checks for time gaps of equal length between sessions. If there are four consecutive sessions with equal time gaps between them and they are followed by a fifth session with the same time gap, then the algorithm flags the fifth session as a scripted session.

    Regarding the size of time gaps and how big a gap qualifies as a time gap worth monitoring, the algorithm considers the time elapsed between two sessions to be a time gap if the length of the gap is equal to or greater than 10 minutes and equal to or less than two days.

The range of algorithms available is planned to be extended in future releases.

SPA automatically runs an algorithm evaluator tool each day to evaluate how well these algorithms for analytics are working on the current dataset residing on the SPS deployment. For more information on this tool, see Algorithm evaluation. If you want more information on how to interpret the evaluation results on your SPS deployment, contact our Support Team.

Algorithm evaluation

The algorithm evaluator tool is a support tool used to evaluate how well the machine learning algorithms for SPS analytics are working on the current dataset residing in the SPS deployment. The tool is run every day automatically by `analytics-daily.service`, but you can also run it manually by issuing the following commands on the console, when instructed to do so by One Identity Support:

1. `make-cross-scores`: This command performs a scoring calculation that serves as a basis for the algorithm evaluation procedure.
2. `evaluate-cross-scores`: This command evaluates the metrics of the scoring performed with the `make-cross-scores` command, and generates the report of the evaluation, available at the following location:

   `/opt/pam-pipeline/var/algoeval-report/`

The report directory contains the evaluation results in a `report.txt` file, and in several plot image files in a `*.png` format. Send these report files to One Identity Support when instructed.

Enable One Identity Safeguard for Privileged Analytics

Prerequisites:

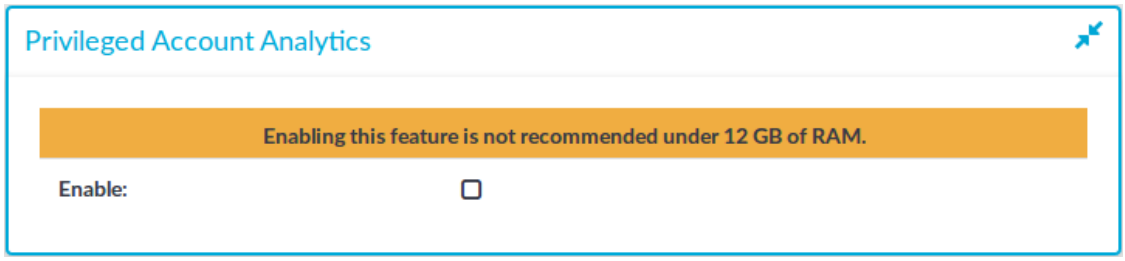
A license that has One Identity Safeguard for Privileged Analytics (SPA) enabled.

The following describes how to enable SPA.

To enable SPA

1. Go to **Basic Settings > Local Services > Privileged Account Analytics**.

   Figure 1: Basic Settings > Local Services > Privileged Account Analytics

2. Select the **Enable** checkbox.
3. Click Commit.

5

Configure analytics

Select the analytics (also called algorithms) that you wish to use to analyze session data and enable them in SPS.

Session data is scored by any combination of algorithms that you enable. The scores given by algorithms are aggregated to create a single score.

During the aggregation process, the lowest and highest scores are removed. This is required in order to lower the number of false positives and false negatives. A typical attack is indicated by signs of unusual user behavior, unusual from multiple points of view. However, some things are usually perfectly normal about even the strangest sessions. This is why removing scores at the two extremes helps minimize the number of false positives and false negatives.

The following describes how to configure algorithms in SPS.

To configure algorithms in SPS

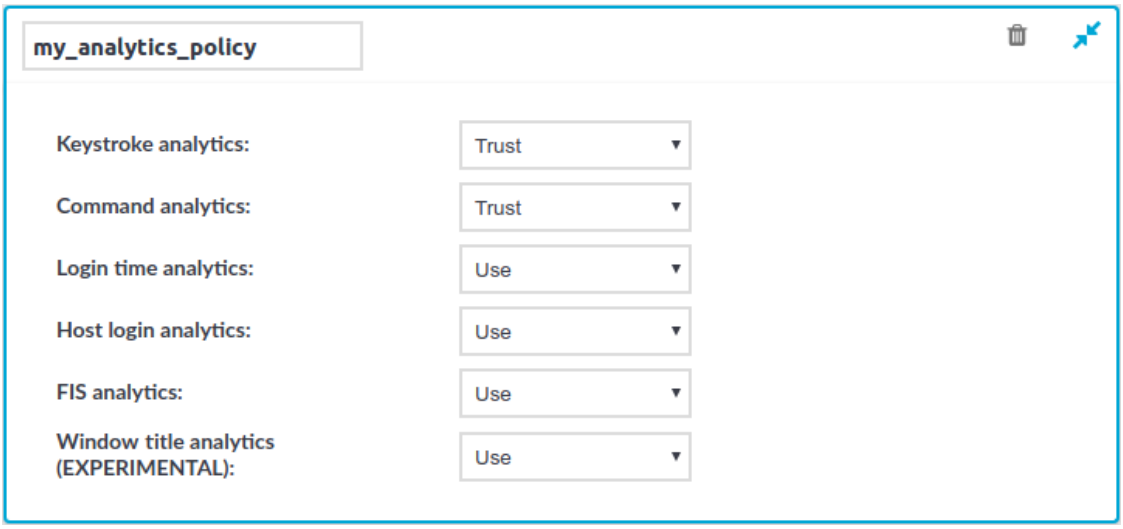
1. In SPS, go to **Policies > Analytics Policies**.
2. Enter a name for your analytics policy.
3. For each algorithm, select one of the following values:
   - **Disable**: Select this value if you do not want to use a particular algorithm.
   - **Use**: Select this value if you want to use a particular algorithm.
   - **Trust**: Select this value if you want to use a particular algorithm, and wish to include in the final aggregated score all the scores given by this algorithm.

     Remember that during score aggregation, the lowest and highest scores are removed. You can choose to override this principle by selecting **Trust** for those algorithms that you wish to have a bigger weight in the final, aggregated, single score.
4. The **Scripted session detection** option is enabled by default. Decide whether or not you want to enable the detection of scripted sessions.

   Scripted session detection is currently done by the clockmaster and gapminder algorithms.
5. Click 

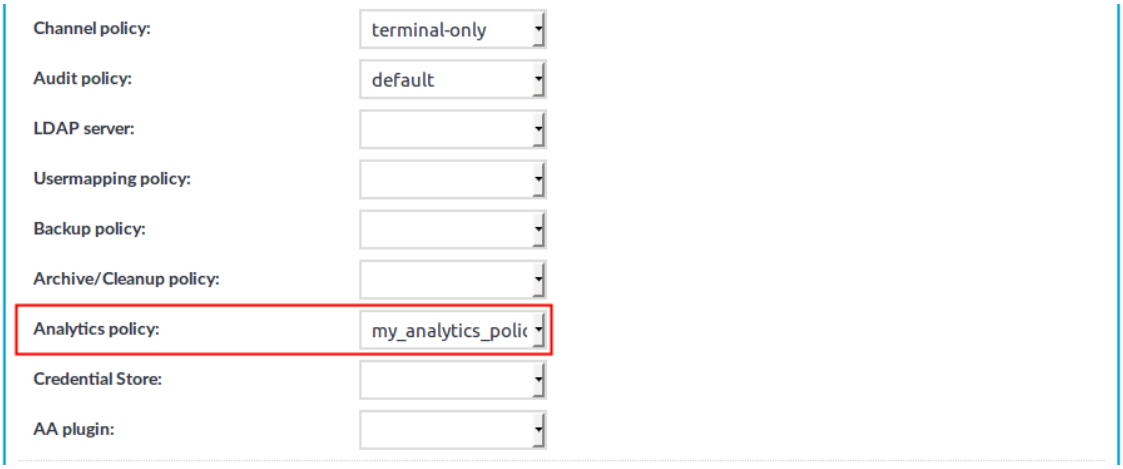
.

Figure 2: Policies > Analytics Policies — Recording an analytics policy



6. Go to **<Protocol name> Control > Connections**, and in the **Analytics policy** field, select the policy that you want to use for your connections.

Figure 3: <Protocol name> Control > Connections > Analytics policy — Adding an analytics policy to a connection policy



7. Click 

.

Analyze data using One Identity Safeguard for Privileged Analytics

Prerequisites

Make sure that you have session data from network traffic that:

- contains real, unique usernames linked to users other than root/administrator or a shared account

  To check this, navigate to **Search**, and check whether the **Username** column contains data. This is important, because session data will be linked to users.

  If you do not have unique usernames in your session data, review your authentication settings and consult with the One Identity Professional Services team to learn about your options to tie accounts to users.
- has commands extracted (using lightweight or full indexing, or in real-time through content policies)

  For instructions on how to configure indexing and include commands in the scope of indexing, see "Indexing audit trails" in the Administration Guide.

  For details on how to configure real-time command extraction using a content policy, see "Creating a new content policy" in the Administration Guide.
- has keystrokes extracted (using lightweight or full indexing, or in real-time through content policies)

  The minimum required amount of data for reliable insight is 5 sessions with approximately 200 keystrokes each.

  For instructions on how to configure indexing and include typing biometrics in the scope of indexing, see "Indexing audit trails" in the Administration Guide.

  For details on how to configure real-time extraction of keystroke-related data using a content policy, see "Creating a new content policy" in the Administration Guide.
- has pointing device (mouse) biometrics extracted (using lightweight or full indexing, or in real-time through content policies)

  For instructions on how to configure indexing and include pointing device biometrics in the scope of indexing, see "Indexing audit trails" in the Administration Guide.

For details on how to configure real-time extraction of pointing device-related data using a content policy, see "Creating a new content policy" in the Administration Guide.

- has window titles extracted (using lightweight or full indexing, or in real-time through content policies)

  For instructions on how to configure indexing and include window titles in the scope of indexing, see "Indexing audit trails" in the Administration Guide.

  For details on how to configure real-time window title extraction using a content policy, see "Creating a new content policy" in the Administration Guide.

The following describes how to start using SPA.

To start using SPA

1. **Build the first baseline.**

   You can do this in either of the following ways:

   - Wait for the daily periodic building to kick in (run by cron).
   - Log in through SSH into the core-shell, and then issue `/opt/pam-pipeline/bin/build-baselines`.

     For the first run, running it with `-ld` (log to console and debug) is a good idea, it will not affect performance.

     It might take a while for baseline building to complete if there is a lot of data, so it makes sense to run it in screen from a fixed node.

   Baseline building is a periodical asynchronous process, meaning that incoming sessions do not immediately change the baseline.

2. **Check whether the baselines have been built.**

   a. Connect to PostgreSQL, the database that permanently stores the outcome of any analyses performed by SPA for later display on the Search interface:

   ```
   psql -U paa paa
   ```

   b. Issue SQL queries to get a list of users for whom a baseline has been built:

   - For the keystroke algorithm, use:

     ```
     select distinct user_id from keystroke;
     ```

   - For the mouse algorithm, use:

     ```
     select distinct user_id from mouse;
     ```

   - For the command algorithm, use:

```
select distinct user_id from command;
```

- For the login time algorithm, use:

  ```
  select distinct user_id from logintime;
  ```

- For the host login algorithm, use:

  ```
  select distinct user_id from hostlogin;
  ```

- For the fis algorithm, use:

  ```
  select distinct user_id from fis;
  ```

- For the window title algorithm, use:

  ```
  select distinct user_id from windowtitle;
  ```

- For the clockmaster algorithm, use:

  ```
  select distinct user_id from clockmaster;
  ```

To generate scores after a successful baseline build, restart the pipeline by entering the following command: `systemctl restart pam-pipeline`.

3. **Start getting scores.**

   Scoring happens in real-time, meaning that as soon as new data (even data from an ongoing session) is available, SPA immediately scores it.

   > TIP: When data is not immediately available to you and you are unable to wait until sufficient amount of data comes in from production traffic, you can resort to manually reindexing historical sessions. For details, see "Reindex historical sessions" in the Safeguard for Privileged Analytics Configuration Guide.

   Scores represent an aggregated amount. Session data is scored by multiple algorithms independent from each other. Scores given by individual algorithms are aggregated to create a single score.

4. **Search for sessions with high scores.**

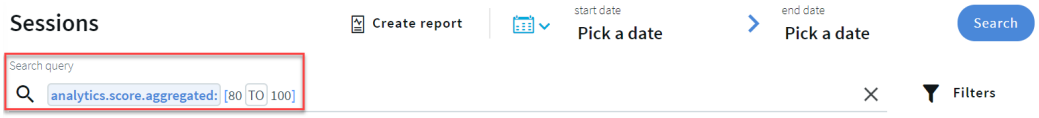
   a. Go to **Search**.

      Sessions are displayed sorted by date. For ongoing sessions, the Search interface is updated in real-time to always show the most up-to-date information.

   b. In the **Search query** field, type `analytics.score.aggregated: [80 TO 100]`, and click **Search**.

      A score between 80 and 100 indicates unusual user behavior.

Figure 4: Searching for sessions with unusual user behavior using a search query



Results that show sessions with high scores are displayed.

Figure 5: Sessions with high scores — table view

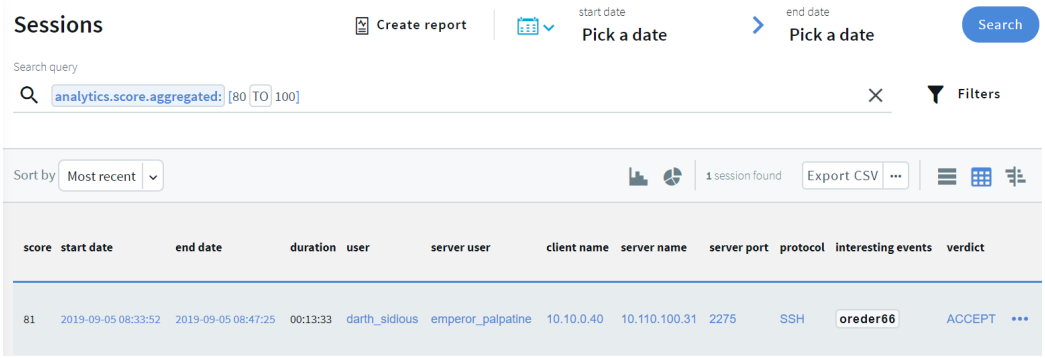
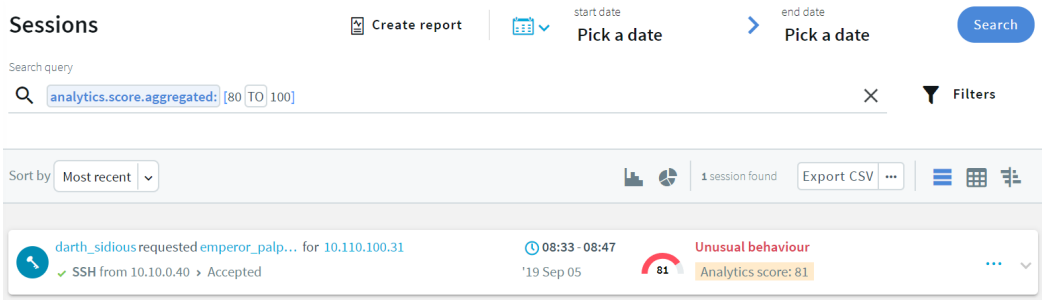


Figure 6: Sessions with high scores — card view



For detailed instructions on how to search effectively and replay audit trails that contain interesting events, see "Using the Search interface" in the Administration Guide.

5. **Alternatively, search for scripted sessions.**

   In the **Search query** field, type `analytics.scripted:true`, and click **Search**.

6. **View the details of a session.**

   To view details of a session, click ••• .

7. **View session analytics.**

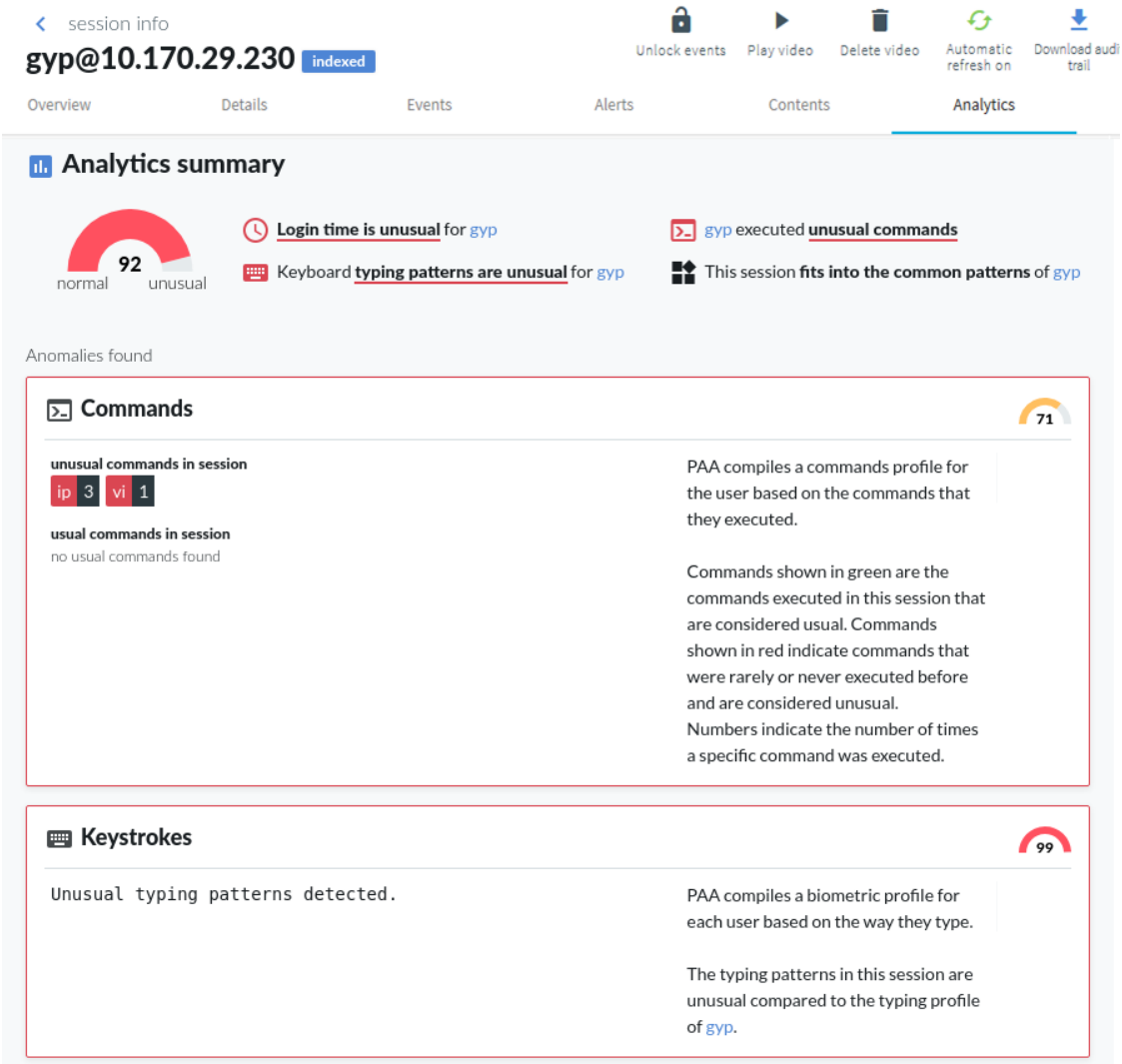
   Click the **Analytics** tab.

   The top of the page displays a summary of key insights about the session, such as:

- The aggregated score (indicated by a gauge). The following color codes are used:
  - Scores between 80-100 indicate unusual behavior, their color code is red.
  - Scores between 70-79 indicate behavior that might require further analysis and attention, their color code is amber.
  - Scores between 0-69 indicate normal behavior, their color code is gray.
- A one-sentence summary of each algorithm's verdict about the session and user behavior.

The **Anomalies found** and **Normal behavior** sections of the page display detailed analyses provided by each of the configured algorithms. This includes short information on how a particular algorithm works and how to read the visualized insight, as well as scores given by the individual algorithms.

Figure 7: Search — Viewing details on the Analytics tab: Anomalies found



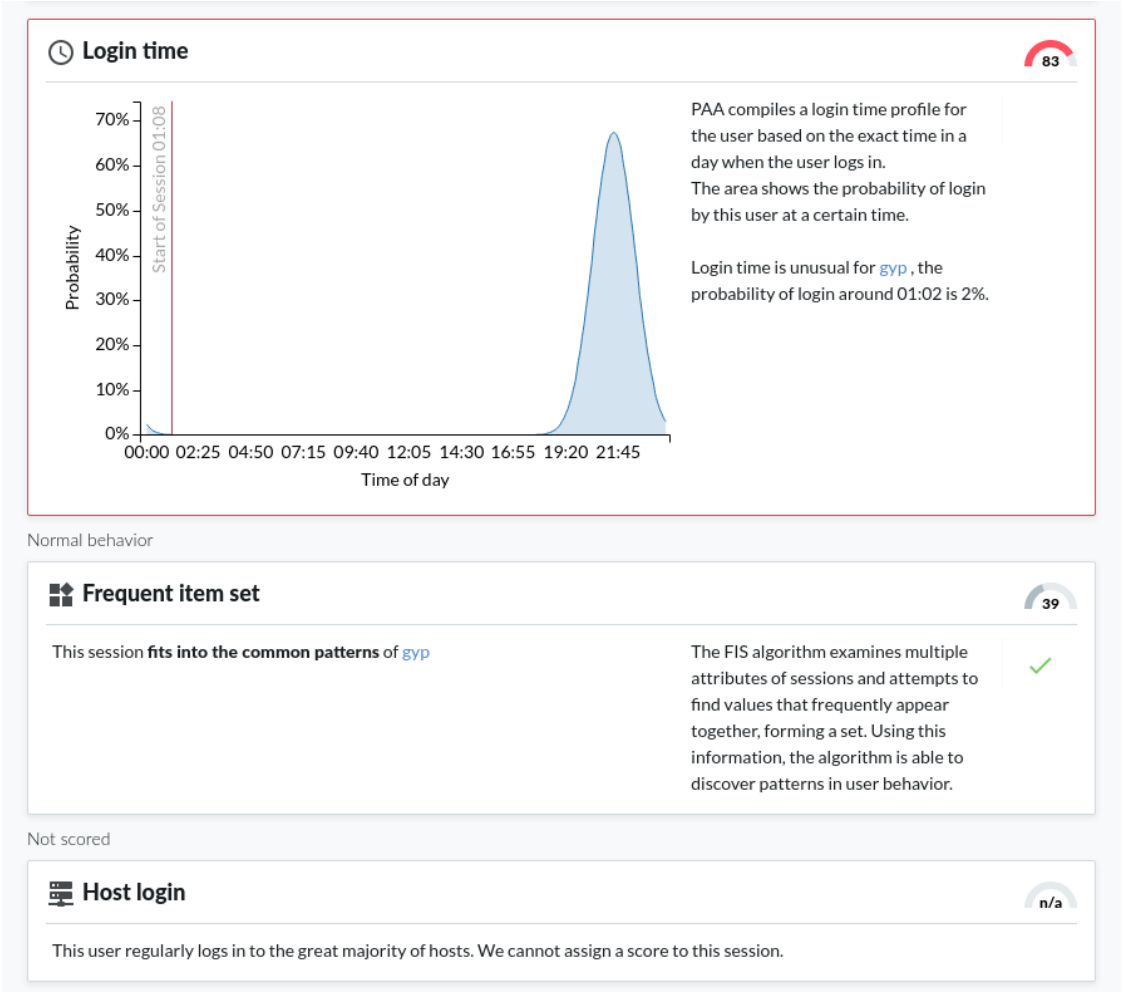
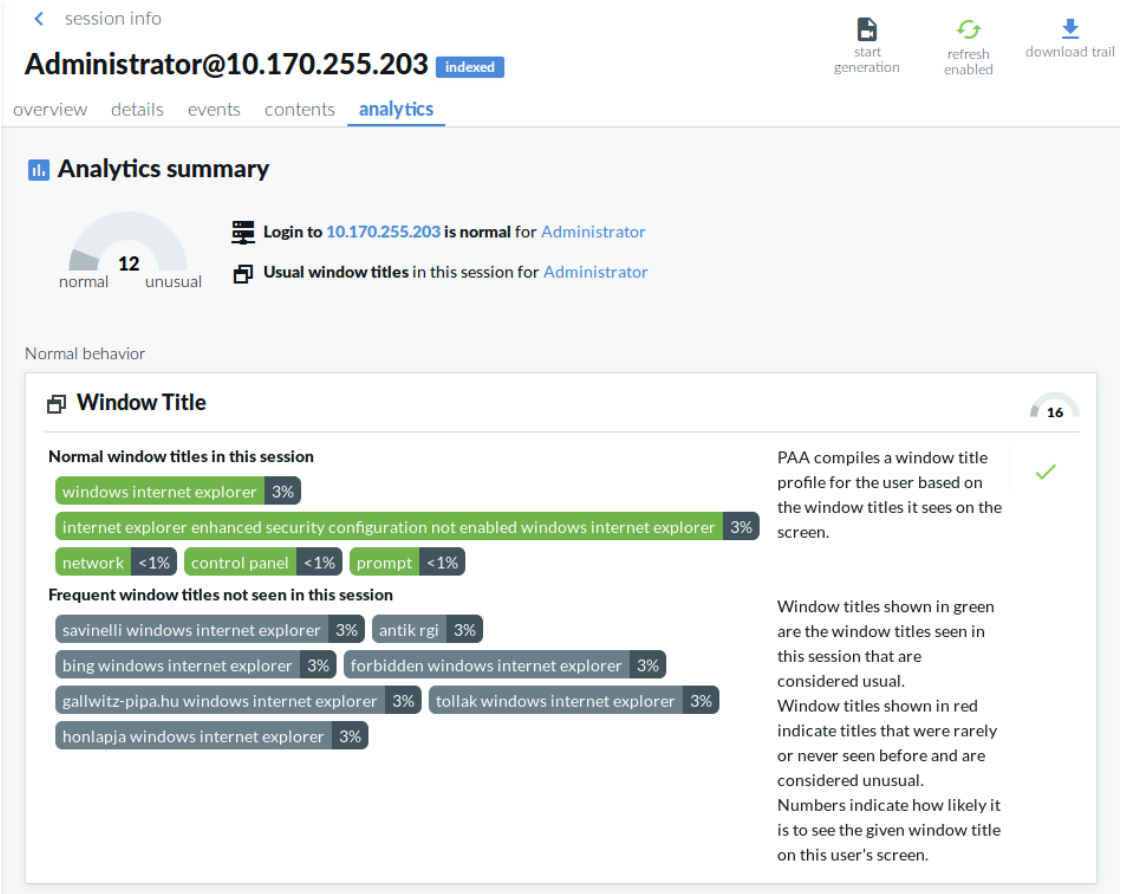
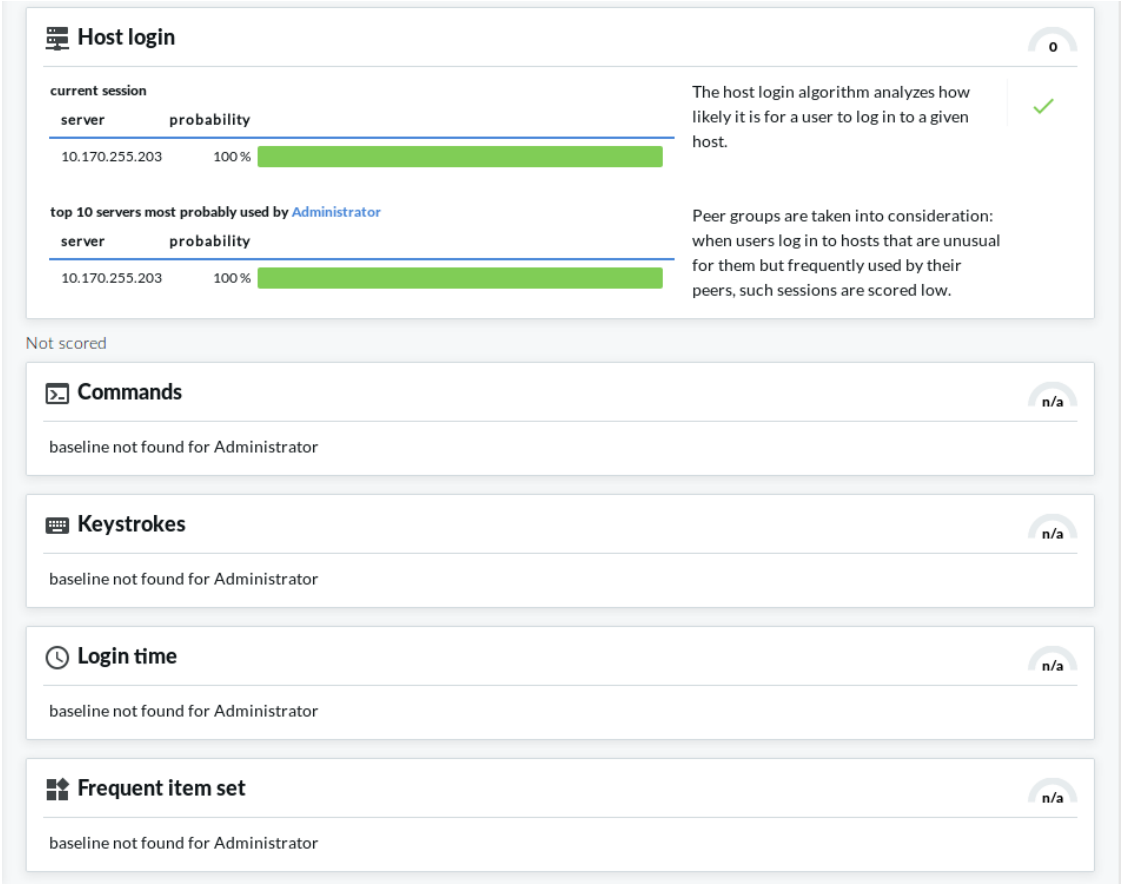

Login time
83
70%
60%
50%
40%
30%
20%
10%
0%
Probability
Start of Session 01:08
00:00 02:25 04:50 07:15 09:40 12:05 14:30 16:55 19:20 21:45
Time of day
PAA compiles a login time profile for the user based on the exact time in a day when the user logs in.
The area shows the probability of login by this user at a certain time.
Login time is unusual for gyp , the probability of login around 01:02 is 2%.
Normal behavior
Frequent item set
39
This session fits into the common patterns of gyp
The FIS algorithm examines multiple attributes of sessions and attempts to find values that frequently appear together, forming a set. Using this information, the algorithm is able to discover patterns in user behavior.
Not scored
Host login
n/a
This user regularly logs in to the great majority of hosts. We cannot assign a score to this session.

Figure 8: Search — Viewing details on the Analytics tab: Normal behavior




Host login
0
current session
server
probability
10.170.255.203
100 %
The host login algorithm analyzes how likely it is for a user to log in to a given host.
top 10 servers most probably used by Administrator
server
probability
10.170.255.203
100 %
Peer groups are taken into consideration: when users log in to hosts that are unusual for them but frequently used by their peers, such sessions are scored low.
Not scored
Commands
n/a
baseline not found for Administrator
Keystrokes
n/a
baseline not found for Administrator
Login time
n/a
baseline not found for Administrator
Frequent item set
n/a
baseline not found for Administrator

7

Reindex historical sessions

When data is not immediately available to you and you are unable to wait until sufficient amount of data comes in from production traffic, you can resort to manual reindexing. For more information, contact our Professional Services Team.

About us

One Identity solutions eliminate the complexities and time-consuming processes often required to govern identities, manage privileged accounts and control access. Our solutions enhance business agility while addressing your IAM challenges with on-premises, cloud and hybrid environments.

Contacting us

For sales and other inquiries, such as licensing, support, and renewals, visit https://www.oneidentity.com/company/contact-us.aspx.

Technical support resources

Technical support is available to One Identity customers with a valid maintenance contract and customers who have trial versions. You can access the Support Portal at https://support.oneidentity.com/.

The Support Portal provides self-help tools you can use to solve problems quickly and independently, 24 hours a day, 365 days a year. The Support Portal enables you to:

- Submit and manage a Service Request
- View Knowledge Base articles
- Sign up for product notifications
- Download software and technical documentation
- View how-to videos at www.YouTube.com/OneIdentity
- Engage in community discussions
- Chat with support engineers online
- View services to assist you with your product