Contents

Admin Activity Audit Log **1**
1. Overview . . . 1
   Key Features . . . 2
   Compatibility . . . 2
2. Installation . . . 2
   Via Composer . . . 2
   Verify . . . 2
3. Configuration . . . 2
   3.1 General . . . 4
   3.2 Logging Rules . . . 4
   3.3 Login Security . . . 4
   3.4 Retention . . . 4
4. Action Log . . . 5
5. Configuration changes . . . 6
6. Change History on entity pages . . . 7
7. Login Attempts and Active Sessions . . . 9
8. Cron Jobs . . . 10
9. Access Control . . . 10
10. Integration event . . . 11
11. Troubleshooting . . . 11
12. Support & Changelog . . . 11

Admin Activity Audit Log

User Guide
Version: 1.0.1
Compatibility: Adobe Commerce / Magento Open Source 2.4.7 – 2.4.9, PHP 8.1+
Vendor: Webmaster Ramos

---

1. Overview

When several people share access to the admin panel, it becomes hard to answer basic questions: who changed a product price, who disabled a CMS block, who edited a configuration value, and when. Adobe Commerce provides built-in Action Logs for exactly this, but Magento Open Source does not. Admin Activity Log, an extension for Magento Open Source and Adobe Commerce, brings a full audit trail to any 2.4 store.

The extension records every change an administrator makes – both through the admin panel and through the admin REST API – with a field-level before/after diff. Configuration changes are recorded the same way, one before/after pair per setting. You can review the log in a dedicated grid, open any record to see the precise fields that changed, watch admin login attempts, and terminate live admin sessions. A retention cron keeps the log within your data-minimization policy.

The extension is read-only by design: it keeps its own journal and never writes to your catalogue, customers, orders or configuration. It is rule-based, deterministic and on-premise – it uses no

external services and sends no data off-site. Sensitive values (passwords, secrets, API keys, tokens, and any pattern you add) are always stored masked.

Key Features

- Field-level audit log for admin-panel and admin REST API changes
- Action Log grid with filters and export, plus a before/after diff page
- Configuration audit: one before/after pair per setting, credentials masked
- Change history on product, customer, category, CMS page, CMS block, order and admin user pages
- Login Attempts grid and Active Sessions control
- Extendable credential masking, retention auto-purge, ACL duty separation

Compatibility

Platform	Versions
Adobe Commerce	2.4.7 – 2.4.9
Magento Open Source	2.4.7 – 2.4.9
PHP	8.1, 8.2, 8.3, 8.4, 8.5
Themes	Not applicable (admin-only)

---

2. Installation

Via Composer

```
composer require webmaster-ramos/module-admin-activity-log
bin/magento module:enable WebRamos_AdminActivityLog
bin/magento setup:upgrade
bin/magento setup:di:compile     # production mode only
bin/magento cache:flush
```

Verify

```
bin/magento module:status WebRamos_AdminActivityLog
```

---

3. Configuration

Navigate to **Stores → Configuration → WebRamos → Admin Activity Log**. All settings are global: an audit configuration must not vary per store.

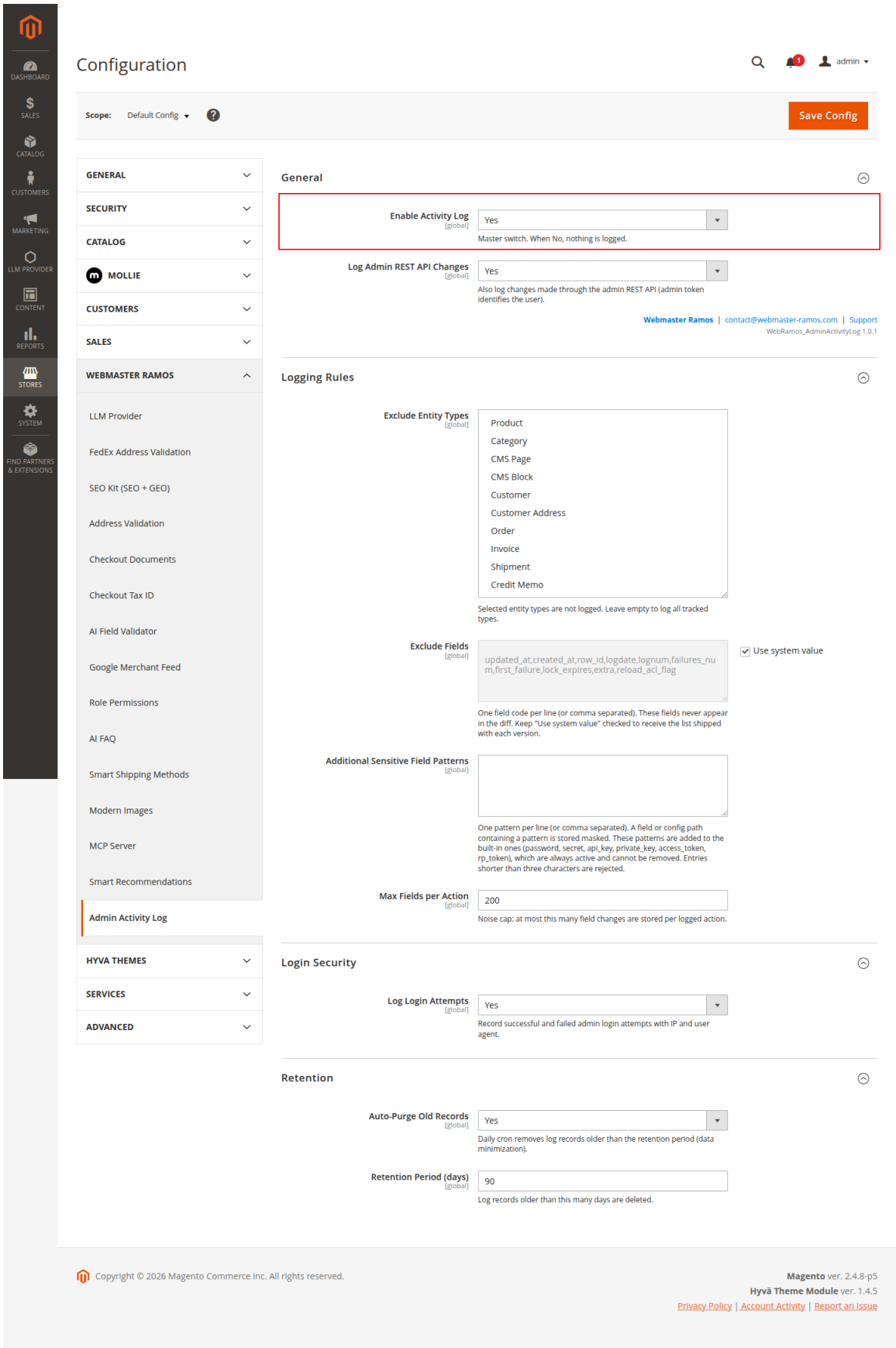


Figure 1: Configuration

3.1 General

Field	Description	Default
Enable Activity Log	Master switch. When set to No, nothing is logged.	No
Log Admin REST API Changes	Also log changes made through the admin REST API. The admin token identifies the acting user.	Yes

The group ends with the vendor banner: the module identifier and the installed version – the string to quote in a support request.

3.2 Logging Rules

Field	Description	Default
Exclude Entity Types	Entity types that are never logged. Leave empty to log all tracked types.	–
Exclude Fields	Field codes (one per line) that never appear in the diff.	updated_at, created_at, row_id, …
Additional Sensitive Field Patterns	Field names and configuration paths containing one of these patterns are stored masked. Adds to the six built-in patterns (password, secret, api_key, private_key, access_token, rp_token), which cannot be switched off. Entries shorter than three characters are rejected.	–
Max Fields per Action	Noise cap: at most this many field changes are stored per record.	200

3.3 Login Security

Field	Description	Default
Log Login Attempts	Record successful and failed admin logins with IP and user agent.	Yes

3.4 Retention

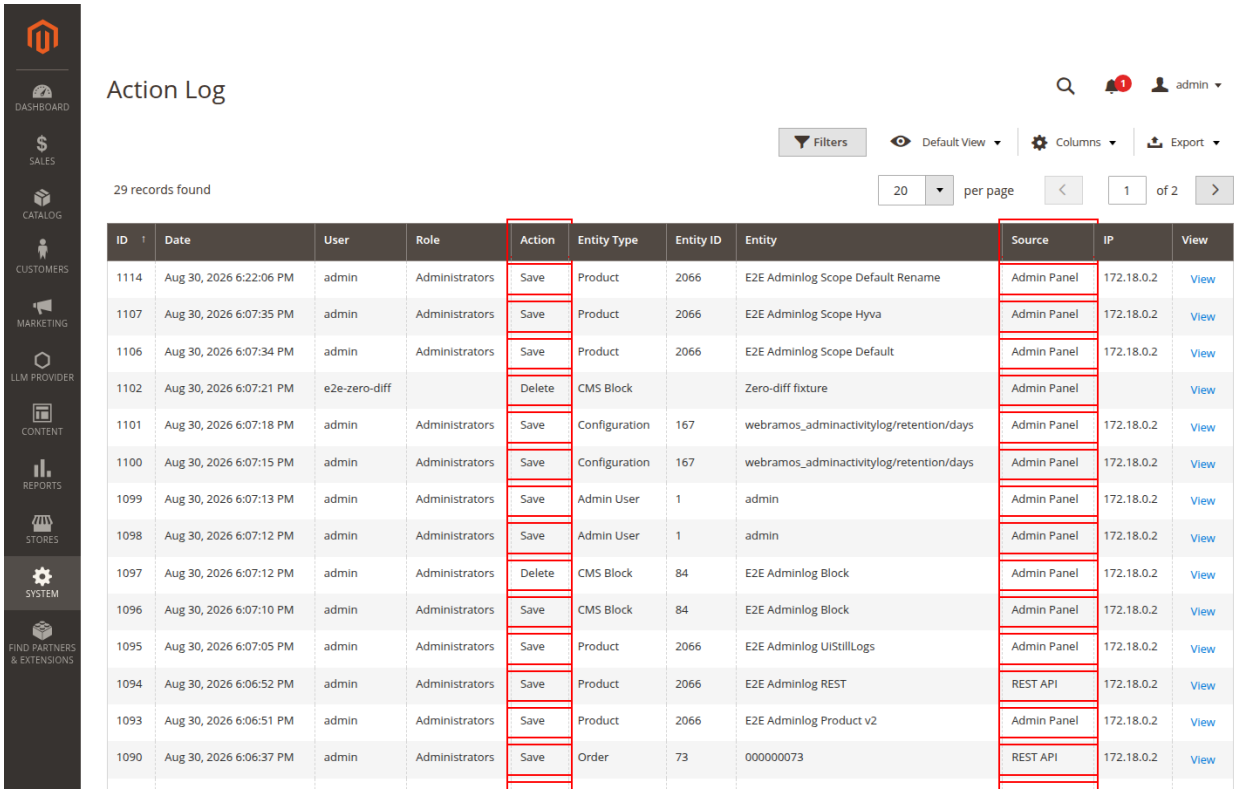
Field	Description	Default
Auto-Purge Old Records	Daily cron removes log records older than the retention period.	Yes
Retention Period (days)	Records older than this many days are deleted.	90

---

4. Action Log

Open **System → Admin Activity Log → Action Log**. The grid lists every recorded change with date, user, role, action type (save / delete), entity type and label, source channel (Admin Panel or REST API) and IP. Use the filters to narrow by user, entity, date or channel, and the Export control to download the current view.

The **Entity** filter matches the record's name as it was **at the moment the change was recorded**, because that is what the journal stores – a renamed record therefore has its earlier entries under its previous name. To see everything ever recorded against one record, filter by **Entity ID** instead: the ID does not change when the name does.



ID	Date	User	Role	Action	Entity Type	Entity ID	Entity	Source	IP	View
1114	Aug 30, 2026 6:22:06 PM	admin	Administrators	Save	Product	2066	E2E Adminlog Scope Default Rename	Admin Panel	172.18.0.2	View
1107	Aug 30, 2026 6:07:35 PM	admin	Administrators	Save	Product	2066	E2E Adminlog Scope Hyva	Admin Panel	172.18.0.2	View
1106	Aug 30, 2026 6:07:34 PM	admin	Administrators	Save	Product	2066	E2E Adminlog Scope Default	Admin Panel	172.18.0.2	View
1102	Aug 30, 2026 6:07:21 PM	e2e-zero-diff		Delete	CMS Block		Zero-diff fixture	Admin Panel		View
1101	Aug 30, 2026 6:07:18 PM	admin	Administrators	Save	Configuration	167	webramos_adminactivitylog/retention/days	Admin Panel	172.18.0.2	View
1100	Aug 30, 2026 6:07:15 PM	admin	Administrators	Save	Configuration	167	webramos_adminactivitylog/retention/days	Admin Panel	172.18.0.2	View
1099	Aug 30, 2026 6:07:13 PM	admin	Administrators	Save	Admin User	1	admin	Admin Panel	172.18.0.2	View
1098	Aug 30, 2026 6:07:12 PM	admin	Administrators	Save	Admin User	1	admin	Admin Panel	172.18.0.2	View
1097	Aug 30, 2026 6:07:12 PM	admin	Administrators	Delete	CMS Block	84	E2E Adminlog Block	Admin Panel	172.18.0.2	View
1096	Aug 30, 2026 6:07:10 PM	admin	Administrators	Save	CMS Block	84	E2E Adminlog Block	Admin Panel	172.18.0.2	View
1095	Aug 30, 2026 6:07:05 PM	admin	Administrators	Save	Product	2066	E2E Adminlog UiStillLogs	Admin Panel	172.18.0.2	View
1094	Aug 30, 2026 6:06:52 PM	admin	Administrators	Save	Product	2066	E2E Adminlog REST	REST API	172.18.0.2	View
1093	Aug 30, 2026 6:06:51 PM	admin	Administrators	Save	Product	2066	E2E Adminlog Product v2	Admin Panel	172.18.0.2	View
1090	Aug 30, 2026 6:06:37 PM	admin	Administrators	Save	Order	73	000000073	REST API	172.18.0.2	View

Figure 2: Action Log grid

Click **View** on any row to open the record. The details page shows the metadata at the top and a field-by-field diff below, old value beside new. Removed values and empty fields are marked explicitly. Values of sensitive fields show as ****** on both sides.

The details page also names the **Store View** the change belongs to – *All Store Views* for a save made in the default scope, the store view's name and code for a save made in one store view, and *Not store-scoped* for records that have no store dimension, such as administrator accounts and global configuration. The Action Log grid has a matching **Store View** column; it is hidden by default to keep the grid readable and can be switched on through the **Columns** control above the grid.

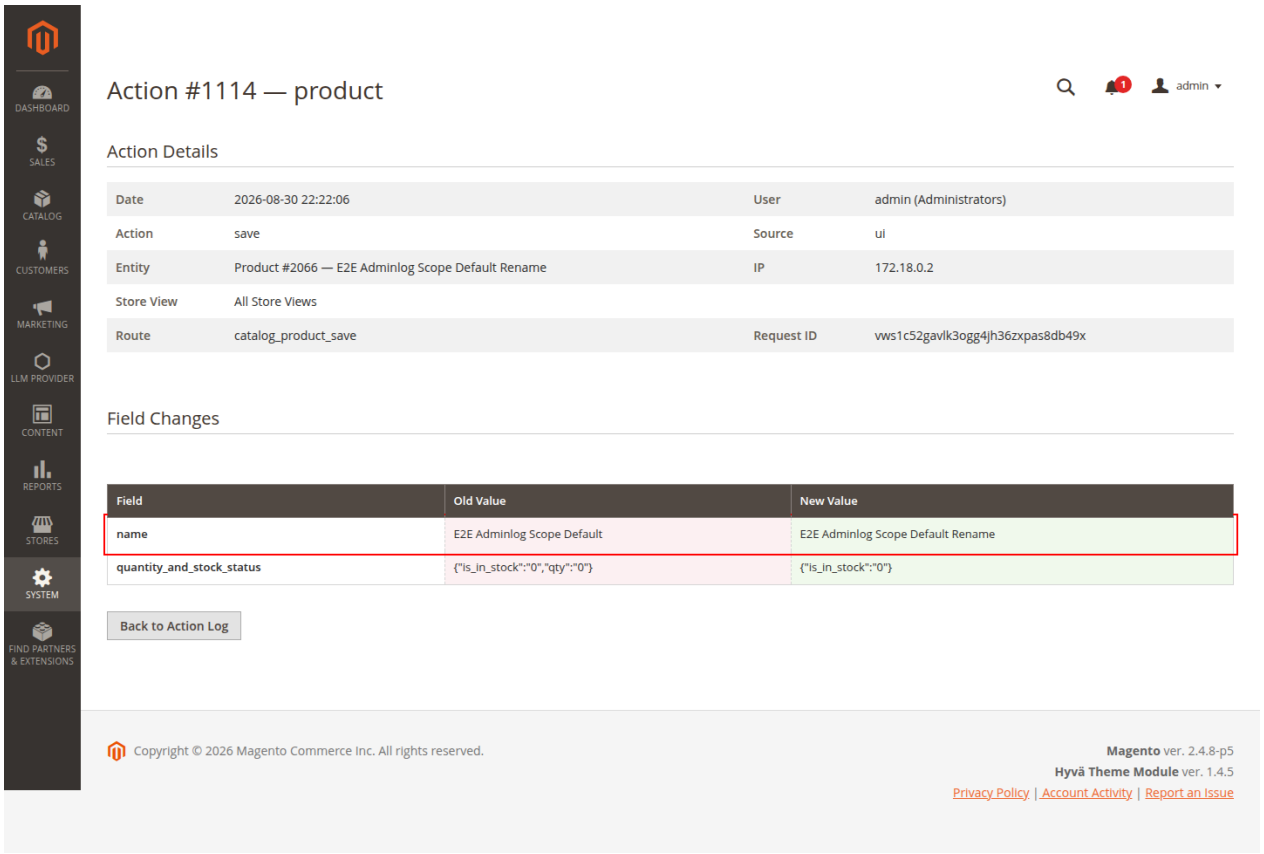


Figure 3: Action diff

---

5. Configuration changes

Every setting saved under **Stores → Configuration** is recorded as its own record with a single row, `value`, old beside new – whether the setting is being set for the first time or changed. A save that changes nothing writes nothing, so saving a section without edits does not fill the log.

Settings stored encrypted (API keys, passwords) and settings whose path matches a sensitive pattern are recorded with ****** on both sides: the log shows that the value changed, never what it was. A change made at website scope carries the website code in the record's label; a change made at store-view scope carries the store view.

---

6. Change History on entity pages

The history of a specific record is shown where you edit it:

- **Products**, **customers**, **categories**, **CMS pages** and **CMS blocks** get a collapsible *Change History* section at the bottom of the edit form.
- **Orders** get a *Change History* tab on the order view. It lists the order's own changes together with the invoices, shipments and credit memos issued against it, each row naming the record it belongs to.
- **Administrators** get a *Change History* tab on the user form, holding two separately labelled sections: *History of this account* – what was changed on that administrator's record, by anyone – and *What this administrator changed* – everything that administrator did elsewhere in the admin.

Each shows the relevant log entries with a link to the full diff, and a record opened from one of them offers a button back to the record it belongs to, beside *Back to Action Log*.

What this administrator changed matches entries by the account's **ID**, not by user name. If an account is deleted and a new one created with the same name, the earlier entries stay in the Action Log under the previous account and are found there by user name.

All of them appear only for roles that hold **View Action Log** (§9). An administrator without it does not see an empty panel – the section is not part of the page at all. A *new* record that has not been saved yet has no history, so the section is empty there by design.

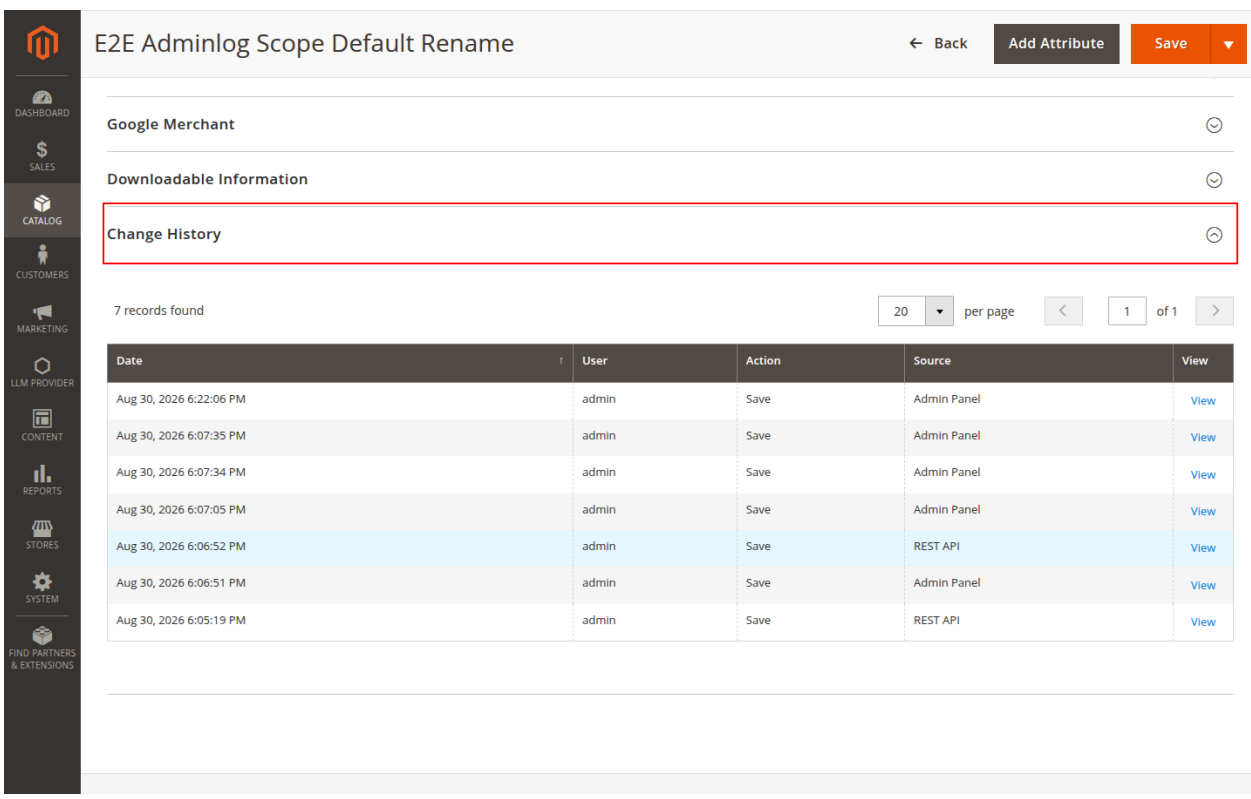


Figure 4: Change History on a product

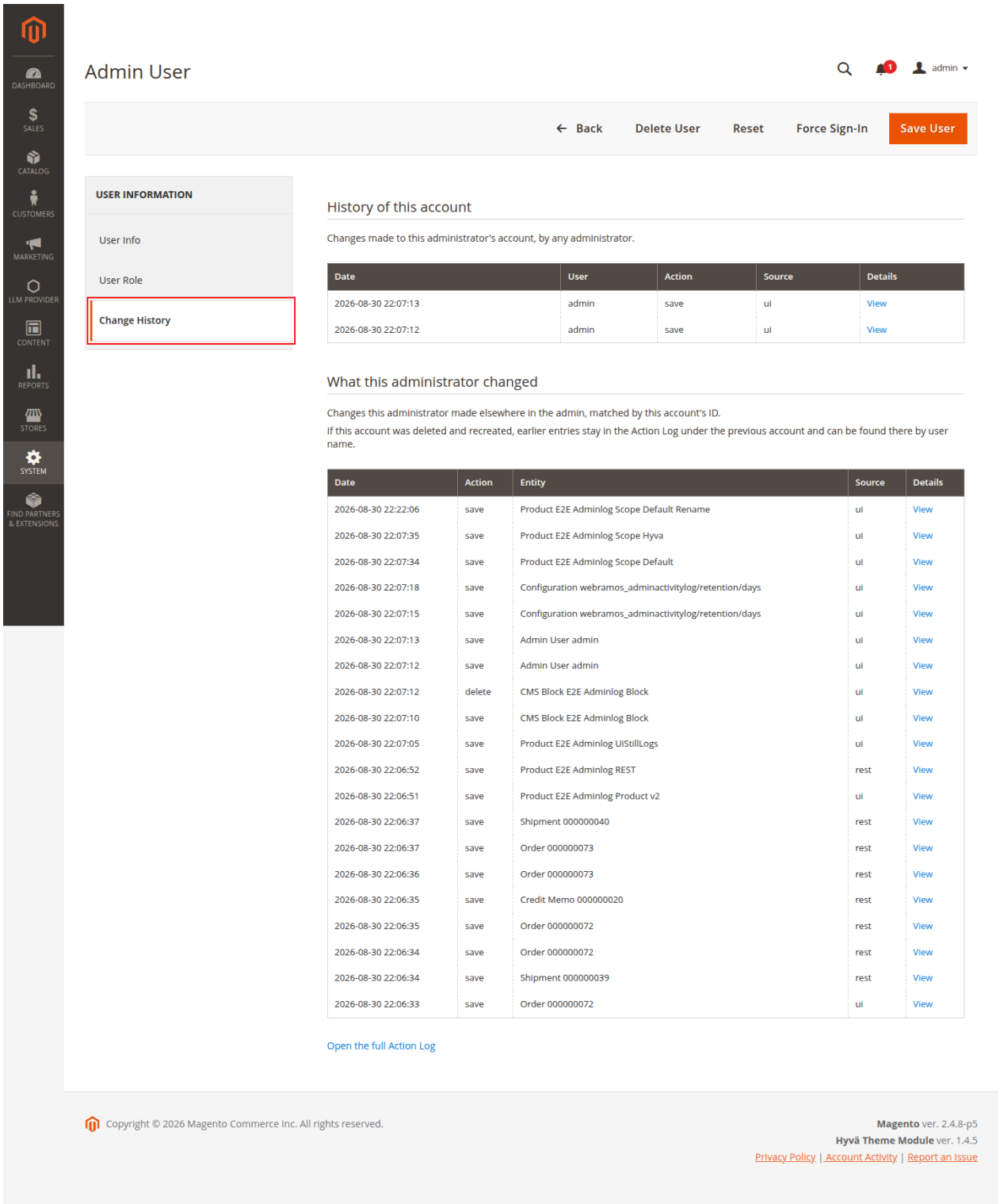
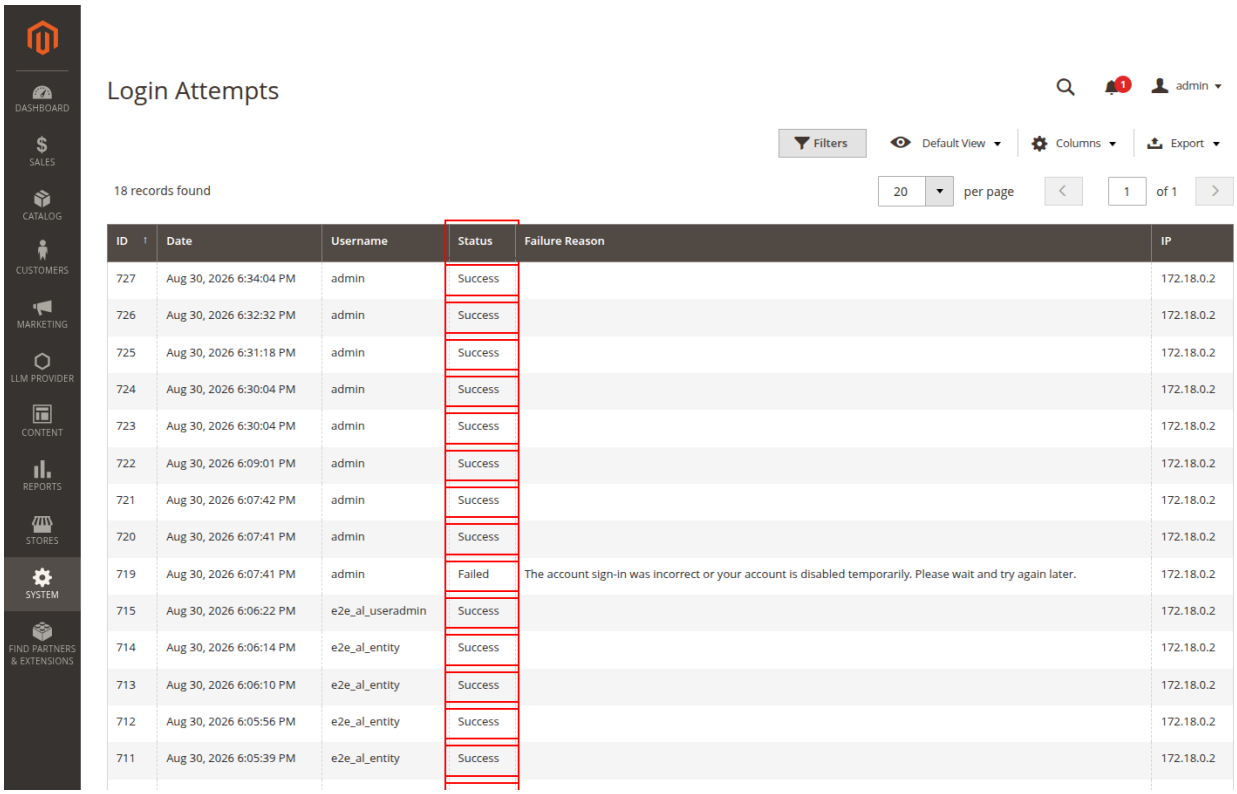


Figure 5: Change History on an administrator

7. Login Attempts and Active Sessions

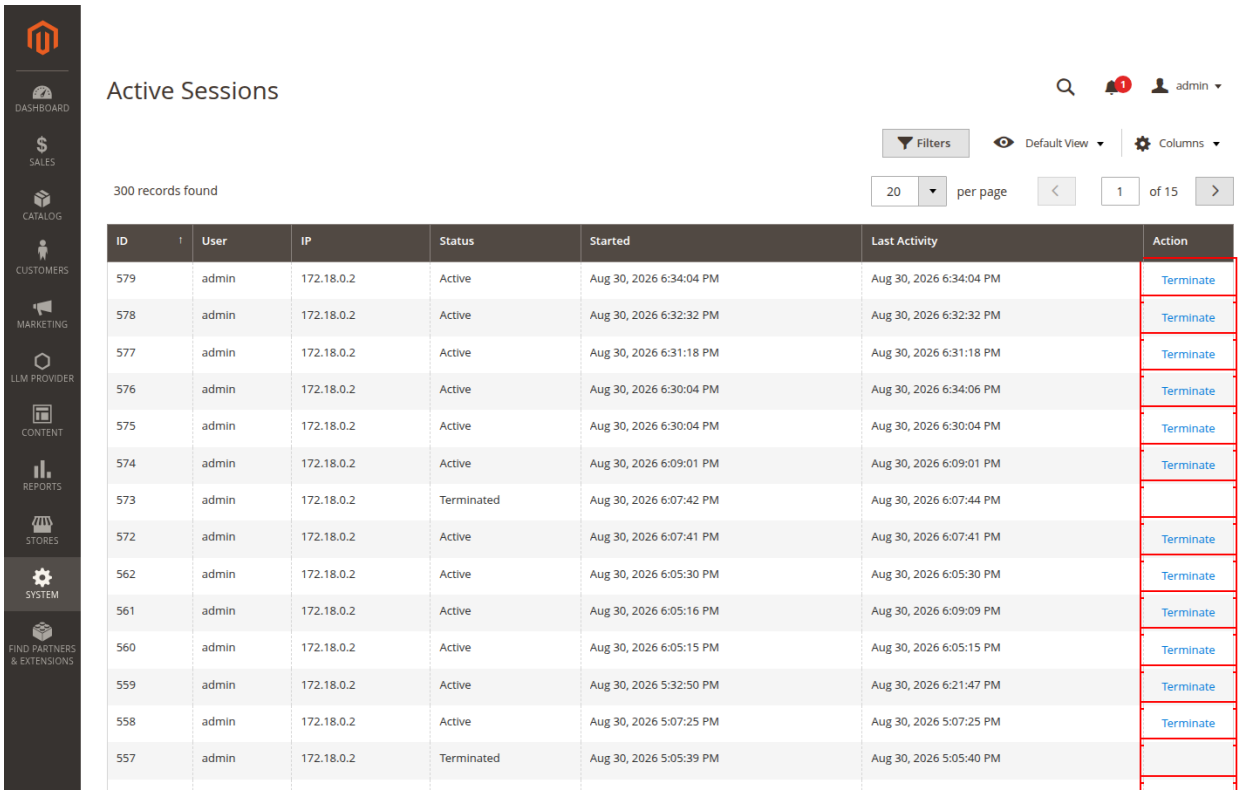
System → Admin Activity Log → Login Attempts lists successful and failed admin logins with username, IP, user agent and, for failures, the reason.



ID	Date	Username	Status	Failure Reason	IP
727	Aug 30, 2026 6:34:04 PM	admin	Success		172.18.0.2
726	Aug 30, 2026 6:32:32 PM	admin	Success		172.18.0.2
725	Aug 30, 2026 6:31:18 PM	admin	Success		172.18.0.2
724	Aug 30, 2026 6:30:04 PM	admin	Success		172.18.0.2
723	Aug 30, 2026 6:30:04 PM	admin	Success		172.18.0.2
722	Aug 30, 2026 6:09:01 PM	admin	Success		172.18.0.2
721	Aug 30, 2026 6:07:42 PM	admin	Success		172.18.0.2
720	Aug 30, 2026 6:07:41 PM	admin	Success		172.18.0.2
719	Aug 30, 2026 6:07:41 PM	admin	Failed	The account sign-in was incorrect or your account is disabled temporarily. Please wait and try again later.	172.18.0.2
715	Aug 30, 2026 6:06:22 PM	e2e_al_useradmin	Success		172.18.0.2
714	Aug 30, 2026 6:06:14 PM	e2e_al_entity	Success		172.18.0.2
713	Aug 30, 2026 6:06:10 PM	e2e_al_entity	Success		172.18.0.2
712	Aug 30, 2026 6:05:56 PM	e2e_al_entity	Success		172.18.0.2
711	Aug 30, 2026 6:05:39 PM	e2e_al_entity	Success		172.18.0.2

Figure 6: Login Attempts

System → Admin Activity Log → Active Sessions lists live admin sessions. With the *Terminate Sessions* ACL resource you can force-log-out any session; the user is returned to the login page on their next request. Ending a session is the only action the extension performs outside its own tables.



ID	User	IP	Status	Started	Last Activity	Action
579	admin	172.18.0.2	Active	Aug 30, 2026 6:34:04 PM	Aug 30, 2026 6:34:04 PM	Terminate
578	admin	172.18.0.2	Active	Aug 30, 2026 6:32:32 PM	Aug 30, 2026 6:32:32 PM	Terminate
577	admin	172.18.0.2	Active	Aug 30, 2026 6:31:18 PM	Aug 30, 2026 6:31:18 PM	Terminate
576	admin	172.18.0.2	Active	Aug 30, 2026 6:30:04 PM	Aug 30, 2026 6:34:06 PM	Terminate
575	admin	172.18.0.2	Active	Aug 30, 2026 6:30:04 PM	Aug 30, 2026 6:30:04 PM	Terminate
574	admin	172.18.0.2	Active	Aug 30, 2026 6:09:01 PM	Aug 30, 2026 6:09:01 PM	Terminate
573	admin	172.18.0.2	Terminated	Aug 30, 2026 6:07:42 PM	Aug 30, 2026 6:07:44 PM	
572	admin	172.18.0.2	Active	Aug 30, 2026 6:07:41 PM	Aug 30, 2026 6:07:41 PM	Terminate
562	admin	172.18.0.2	Active	Aug 30, 2026 6:05:30 PM	Aug 30, 2026 6:05:30 PM	Terminate
561	admin	172.18.0.2	Active	Aug 30, 2026 6:05:16 PM	Aug 30, 2026 6:09:09 PM	Terminate
560	admin	172.18.0.2	Active	Aug 30, 2026 6:05:15 PM	Aug 30, 2026 6:05:15 PM	Terminate
559	admin	172.18.0.2	Active	Aug 30, 2026 5:32:50 PM	Aug 30, 2026 6:21:47 PM	Terminate
558	admin	172.18.0.2	Active	Aug 30, 2026 5:07:25 PM	Aug 30, 2026 5:07:25 PM	Terminate
557	admin	172.18.0.2	Terminated	Aug 30, 2026 5:05:39 PM	Aug 30, 2026 5:05:40 PM	

Figure 7: Active Sessions

---

8. Cron Jobs

Name	Schedule	Purpose
`webramos_adminactivitylog_purge`	Daily, 03:00	Deletes action, change and login records older than the configured retention period. Runs only when both the module and retention are enabled.

---

9. Access Control

Five ACL resources let you split duties across roles. Grant them under **System → Permissions → User Roles → {role} → Role Resources → WebRamos → Admin Activity Log**:

- **View Action Log** – see the Action Log grid and diff pages, and the *Change History* section on product, customer, category, CMS, order and admin user pages (§6)
- **View Login Attempts** – see the Login Attempts grid
- **View Active Sessions** – see the Active Sessions grid

- **Terminate Sessions** – force-terminate a session
- **Admin Activity Log Configuration** – edit the settings

---

10. Integration event

The extension sends no notifications itself. It dispatches one event your project code can observe to send email, Slack messages, or anything else:

- `webramos_adminlog_action_logged` – after a record is persisted

The event payload carries the action record (ids and labels), never raw field values.

---

11. Troubleshooting

- **Nothing is logged.** Confirm *Enable Activity Log* is set to Yes and that the change was made by an admin user on a tracked entity type. Changes made through CLI, cron or data import are not logged in this version.
- **A REST change is not logged.** Confirm *Log Admin REST API Changes* is Yes. Only admin-token requests are logged; integration and customer tokens are ignored by design.
- **A field I expected is missing from the diff.** Check the *Exclude Fields* list and the *Max Fields per Action* cap.
- **A configuration change is missing.** Keep the `config` cache type enabled (as production does): with it disabled, the previous value is read from the database inside the save and the change is recorded as unchanged.
- **Errors appear in the log.** The module never breaks the admin operation it is logging; any storage problem is written to `var/log/system.log` with the prefix `AdminActivityLog`.

---

12. Support & Changelog

- **Author:** Webmaster Ramos (Ruslan Moskalenko)
- **Email:** contact@webmaster-ramos.com
- **Website:** https://webmaster-ramos.com
- **Support:** https://webmaster-ramos.com/support
- **Changelog:** see CHANGELOG.md inside the package