initMAX

Incident Response with Wazuh a DFIR-IRIS

all our microphones are muted

ask your questions in Q&A, not in the Chat

use Chat for discussion, networking or applause

Agenda

1 The Problem

2 The Solution

3 Demonstration

1

The Problem

initMAX

Wazuh is a SIEM, not (quite)

- Security Information and Event Management system
  - Security Information Management (SIM)
    - log collection, normalization, and storage
  - Security Event Management (SEM)
    - real-time event correlation, alerting, and incident response
- Wazuh is very strong on the SIM side, but falls short on the SEM side.

initMAX

Wazuh is a SIEM, not (quite)

1. Limited Real-Time Correlation

- Wazuh supports rules and decoders, but:
  - The correlation is static, rule-based, and lacks the complexity of event relationship modeling.
  - No real temporal correlation (e.g., “If X happens, then Y within 10 minutes, trigger Z”).
  - No support for multi-step attack detection across different log sources or sessions.

Dynamic, behavior-based correlation is not natively supported.

initMAX

Wazuh is a SIEM, not (quite)

2. No Native Event Timeline or Investigation Workflow

- A real Event Management component typically includes:
  - Central incident timeline views.
  - Automated grouping of related alerts into incidents.
  - Analyst investigation workflows
    (e.g., assigning tickets, adding notes, changing status).

You need to build something custom around it — like Kibana, Grafana, or integrate with a separate SOAR platform.

initMAX

Wazuh is a SIEM, not (quite)

3. Weak Incident Lifecycle Management

- Other SIEMs have built-in tools for:
  - Tracking the status of incidents (open, investigating, resolved).
  - Assigning tasks or incidents to people.
    (e.g., analysts, sysadmins, network engineers, …)
  - Enforcing playbooks or automated responses.

You'd need to plug it into something for case management or a custom ticketing system.

initMAX

Wazuh is a SIEM, not (quite)

4. Limited Intelligence Fusion

- Real Event Management tools fuse:
  - Threat intel feeds
  - Behavioral baselines
  - Asset context
    (to enrich events and raise the signal-to-noise ratio)

Wazuh supports tools for threat intelligence (e.g., VirusTotal, MISP, MITRE, CIS, ...), but it's not automated into alert prioritization.

2

The Solution

initMAX

SOAR platform

It’s a class of tools that help automate and manage security operations workflows, especially around incident response.

- Security
- Orchestration
- Automation
- Response

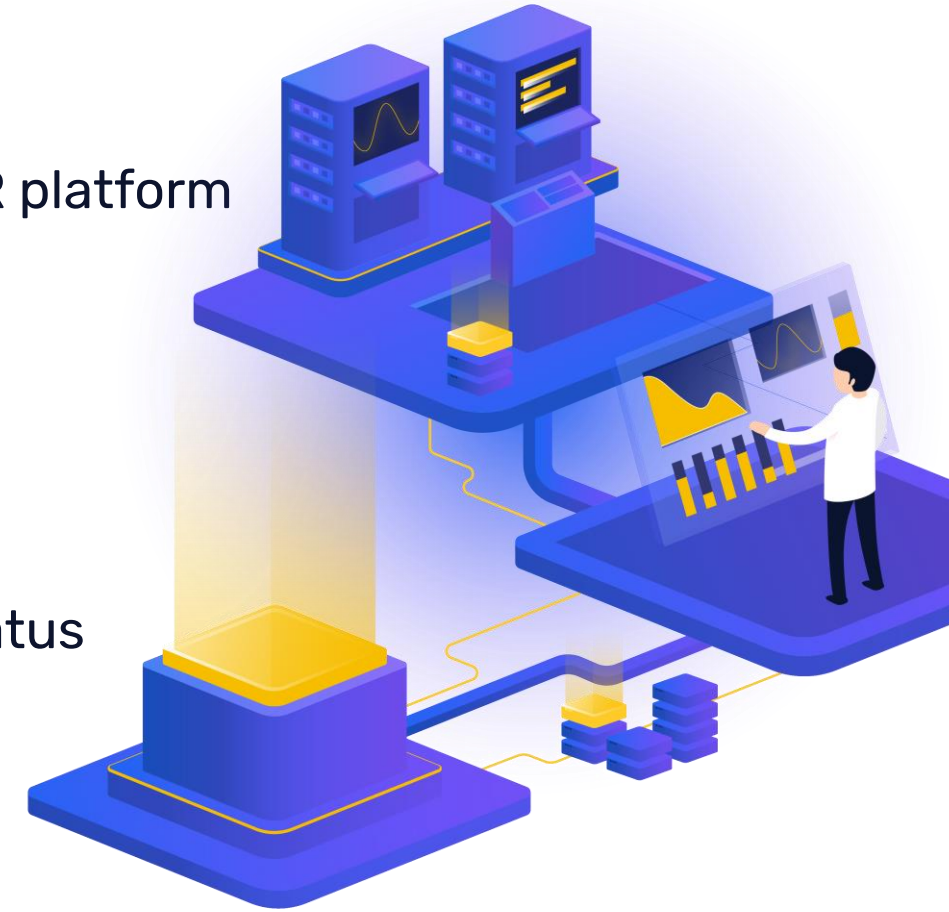
Basically, if SIEM tells you something bad might be happening, SOAR helps you do something about it – quickly, consistently, and automatically.

initMAX

How SOAR complements Wazuh

Wazuh doesn’t have strong incident management, a SOAR platform can act as the "response brain" of your security stack.

- Ingests alerts from Wazuh.
- Enriches the data (e.g., querying threat intel feeds).
- Correlates it with other data sources.
- Manages cases (analyst workflow, documentation, status tracking).
- Automates response actions



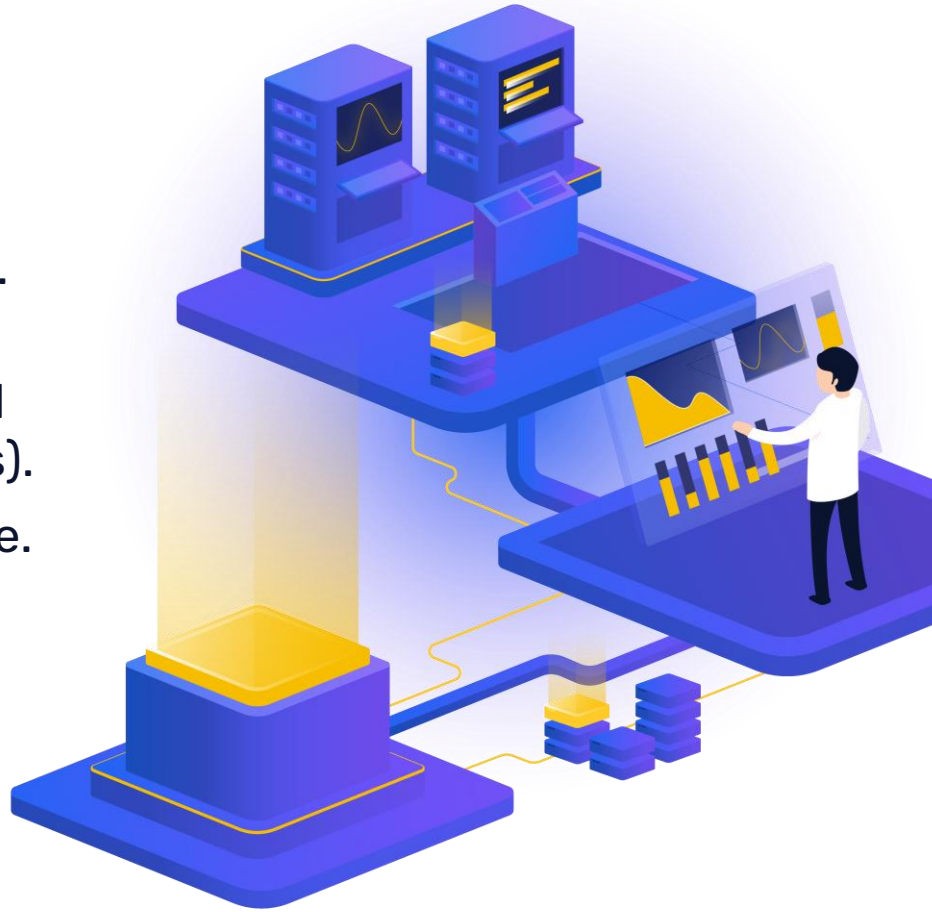
Wazuh → SOAR → Response = Full workflow

initMAX

How SOAR complements Wazuh

Few popular SOAR solutions:

TheHive	Open-source, great for case management. Integrates well with Wazuh via Cortex.
Cortex	Works with TheHive to perform automated response actions (think plugins/playbooks).
Shuffle	Open-source, low-code automation engine. Easy to build custom workflows.
Splunk SOAR	Enterprise-grade, deep integrations. Expensive, but powerful.
IBM Resilient	Strong in regulated environments. Workflow-heavy.
Cortex XSOAR (Palo Alto)	Very feature-rich. Heavy, commercial solution.
Tines	No-code, SaaS-based SOAR. Great UI, easy to use.



initMAX

And than, there’s DFIR-IRIS

Digital Forensics and Incident Response – Incident Response Investigation System is an open-source, robust web application that is designed to:

- Manage incident response cases.
- Document investigations.
- Track artifacts, IOCs, and timeline of events.
- Collaborate with your security team.

It’s like TheHive in some ways, but more focused on DFIR-style incident tracking rather than automation-heavy SOAR features.

initMAX

How DFIR-IRIS complements Wazuh

It fills the "Incident Management" void that Wazuh leaves, though not in an automated fashion like SOAR. It’s more about forensic documentation and workflow support than automatic playbooks.

Wazuh	**DFIR-IRIS**
Collects & analyzes logs	Helps investigate and manage the incident
Generates alerts	Documents findings & tracks progress
Lacks case management	Provides full case & artifact tracking
Static rules + alerting	Human-led investigation & response

initMAX

DFIR-IRIS key features

It is a lightweight alternative to a full SOAR platform.

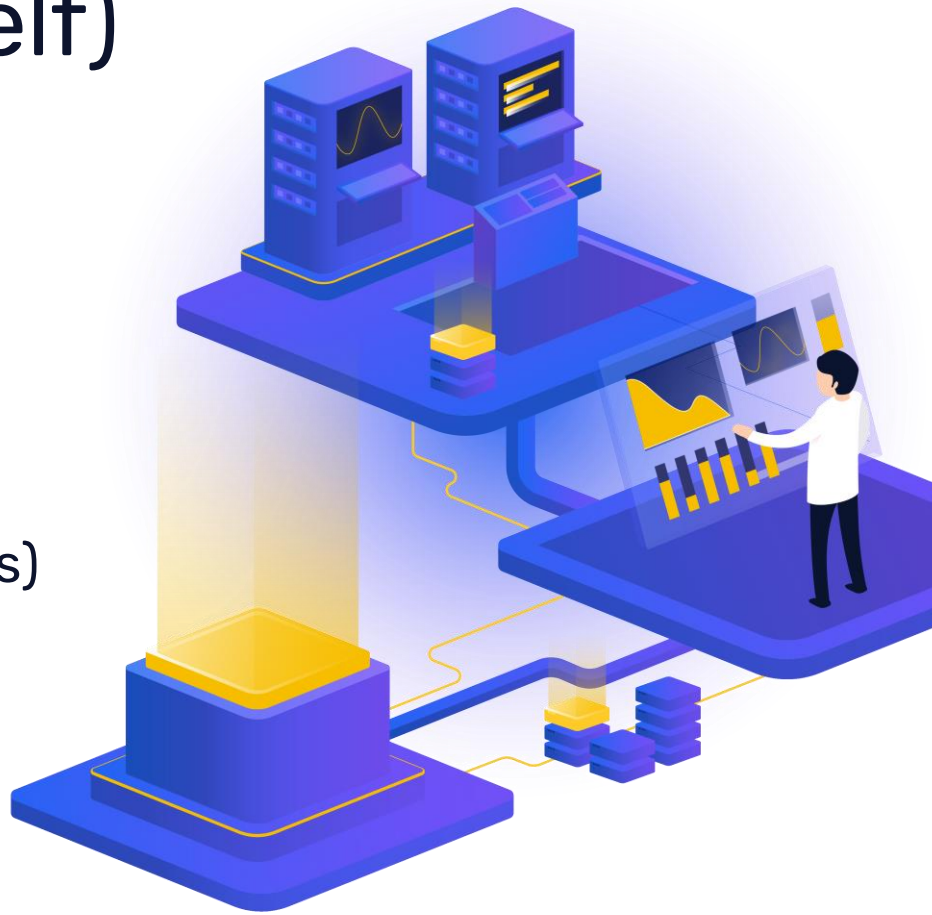
- Case and incident tracking
- Artifact & IOC management
- Timeline of events
- Tagging and severity classification
- Role-based access
- Markdown reporting
- REST API for integrations

initMAX

What DFIR-IRIS doesn't do (by itself)

It is a lightweight alternative to a full SOAR platform.

- No automation (e.g., auto-block IPs, run scripts)
- No built-in correlation engine
- No live alert ingestion (unless you wire it up)
- No response actions (unless integrated with other tools)

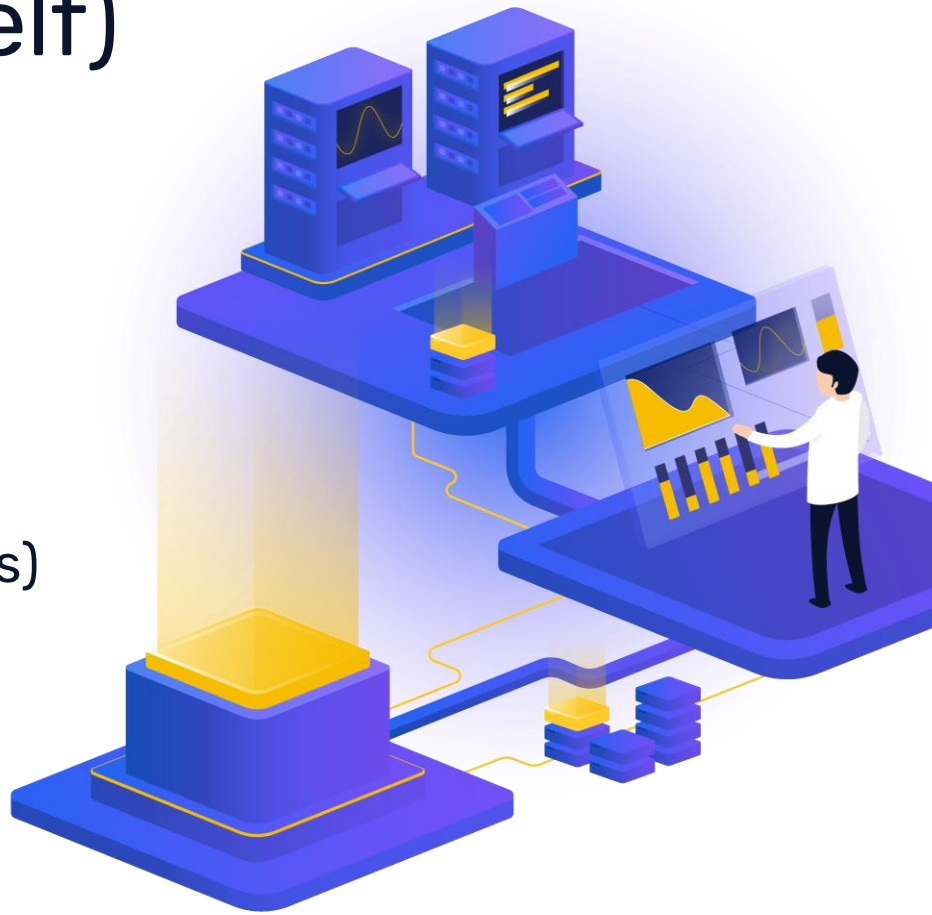


initMAX

What DFIR-IRIS doesn't do (by itself)

It is a lightweight alternative to a full SOAR platform.

- No automation (e.g., auto-block IPs, run scripts)
- No built-in correlation engine
- No live alert ingestion (unless you wire it up)
- No response actions (unless integrated with other tools)



3

Demonstration

initMAX

DFIR-IRIS installation

Installation requirements (docker)

```
dnf config-manager --add-repo https://download.docker.com/linux/rhel/docker-ce.repo
```

```
dnf install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```

```
systemctl enable docker --now
```

initMAX

DFIR-IRIS installation

Installation requirements (git)

```
dnf install git
```

It will (most likely) also upgrade OpenSSL, so do NOT logout of the system before upgrading OpenSSH too.

Otherwise, you might get the following error:

```
OpenSSL version mismatch. Built against 30000070, you have 30200020.
systemd[1]: sshd.service: Main process exited, code=exited, status=255/EXCEPTION
systemd[1]: sshd.service: Failed with result 'exit-code'.
```

initMAX

DFIR-IRIS installation

Clone DFIR-IRIS github repository:

```
git clone https://github.com/dfir-iris/iris-web.git /opt/iris-web/ && cd /opt/iris-web/
```

Checkout the non-beta tagged version:

```
git checkout v2.4.20
```

initMAX

DFIR-IRIS installation

Create environment config file from provided example:

```
cp .env.model .env
```

Modify the environment config file according to your needs:

```
vim .env
```

initMAX

DFIR-IRIS installation

Required changes to the environment config file:

- POSTGRES_PASSWORD (Password of the postgres user)
- POSTGRES_ADMIN_PASSWORD (Password of the db admin user)
- IRIS_SECRET_KEY (Key used by Flask to secure the session cookies)
- IRIS_SECURITY_PASSWORD_SALT (A salt used for password encryption in the DB)
- IRIS_ADM_PASSWORD (Password of the administrator account, policy enforced)
- IRIS_ADM_API_KEY (API key of the administrator, no complexity check)

```
openssl rand -base64 64
```

Make sure the DB password does not contain characters that could be interpreted as an DB url (i.e., # and @)!

initMAX

DFIR-IRIS installation

When changes are made upon running containers, the nginx one needs to be rebuilt:

```
docker-compose stop nginx
docker-compose build nginx --no-cache
docker-compose up
```

When changed beforehand, just start it all up (use –d to run on background):

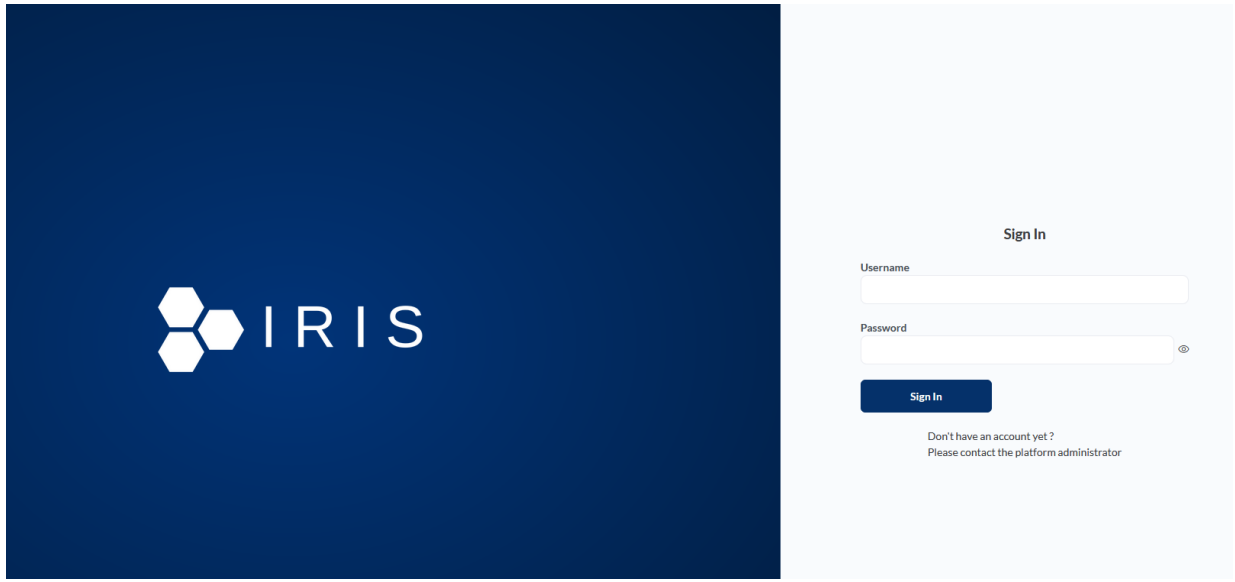
```
docker compose up
```

initMAX

DFIR-IRIS installation

You should now be able to login to the WebUI using chosen password:

```
https://<%IP_ADDRESS%>
```



initMAX

DFIR-IRIS setup

Now, we need to create an automation service account, and a customer.

Create a customer in section **Advanced -> Customers -> Add customer**:

initMAX

DFIR-IRIS setup

Make note of Customer ID, we will need it for integration with Wazuh:

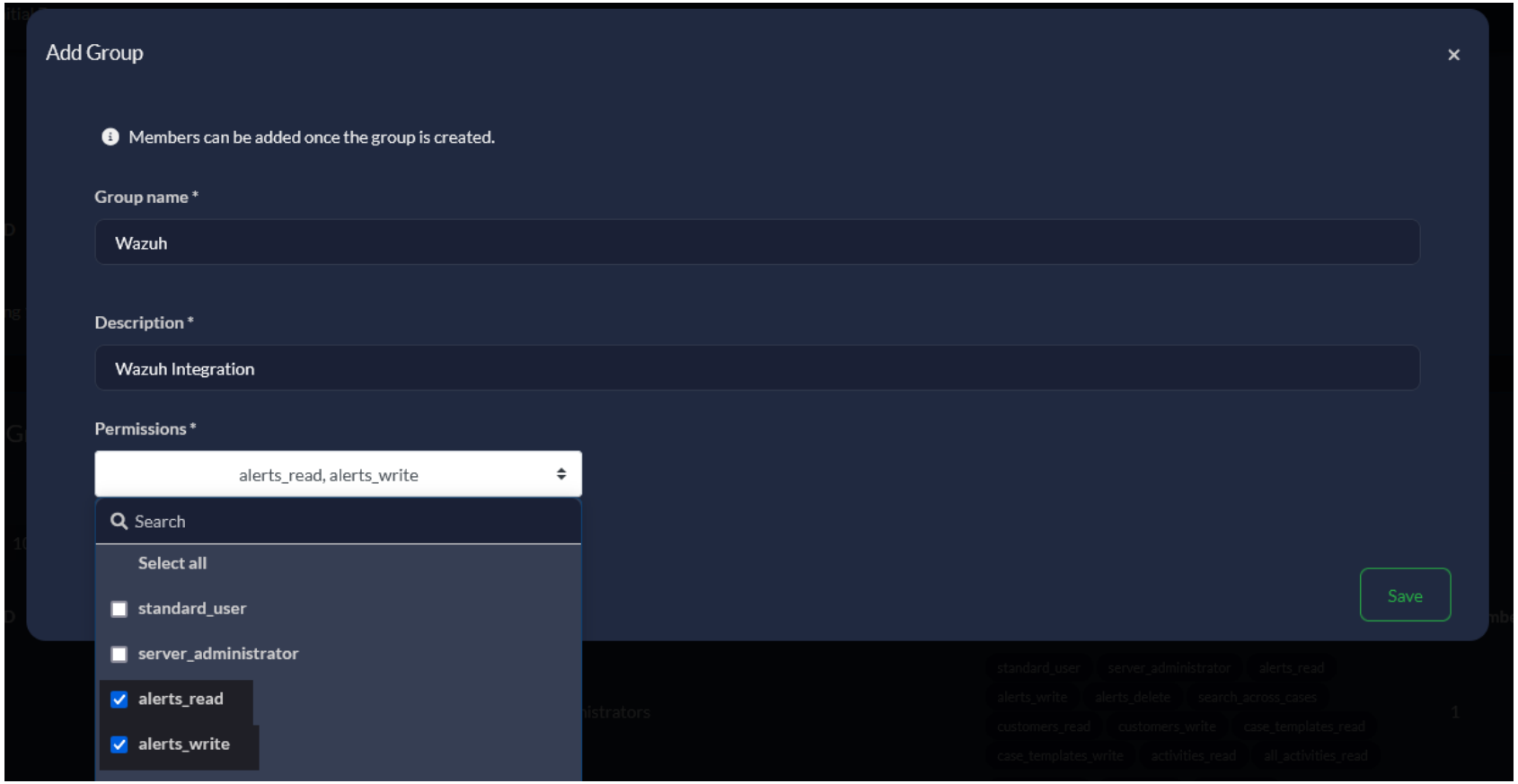
Name	Customer ID
IrisInitialClient	1
initMAX	2

initMAX

DFIR-IRIS setup

Create a permission group in section **Advanced -> Access control -> Add group.**

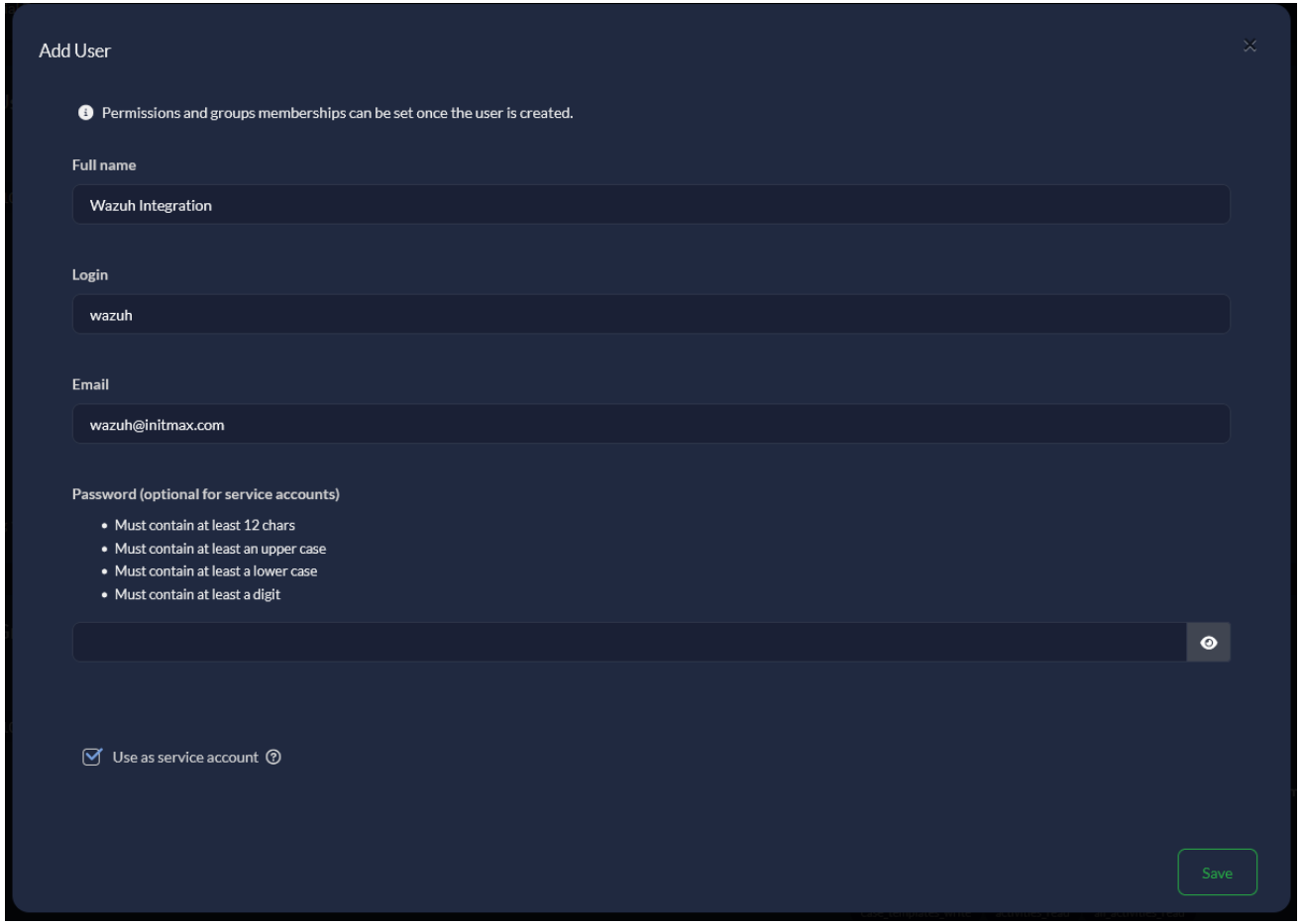
Assign **alerts_read** and **alerts_write** permissions to it.



initMAX

DFIR-IRIS setup

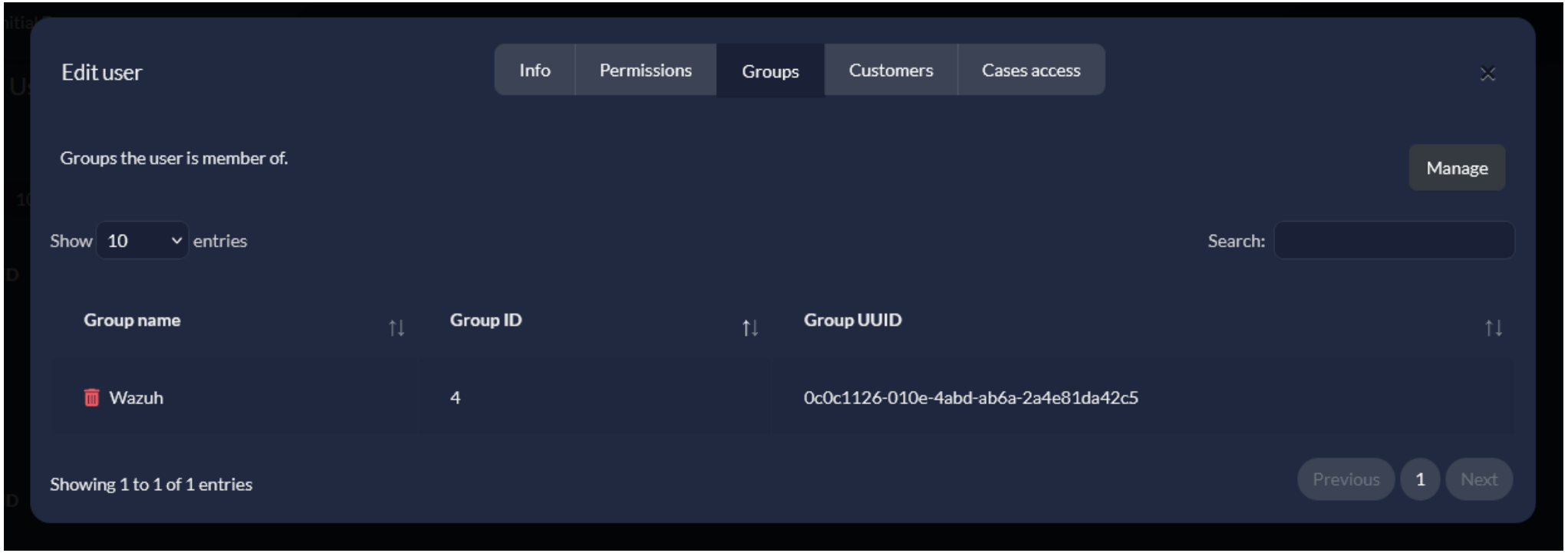
Now create a service account in section **Advanced -> Access control -> Add user**:



initMAX

DFIR-IRIS setup

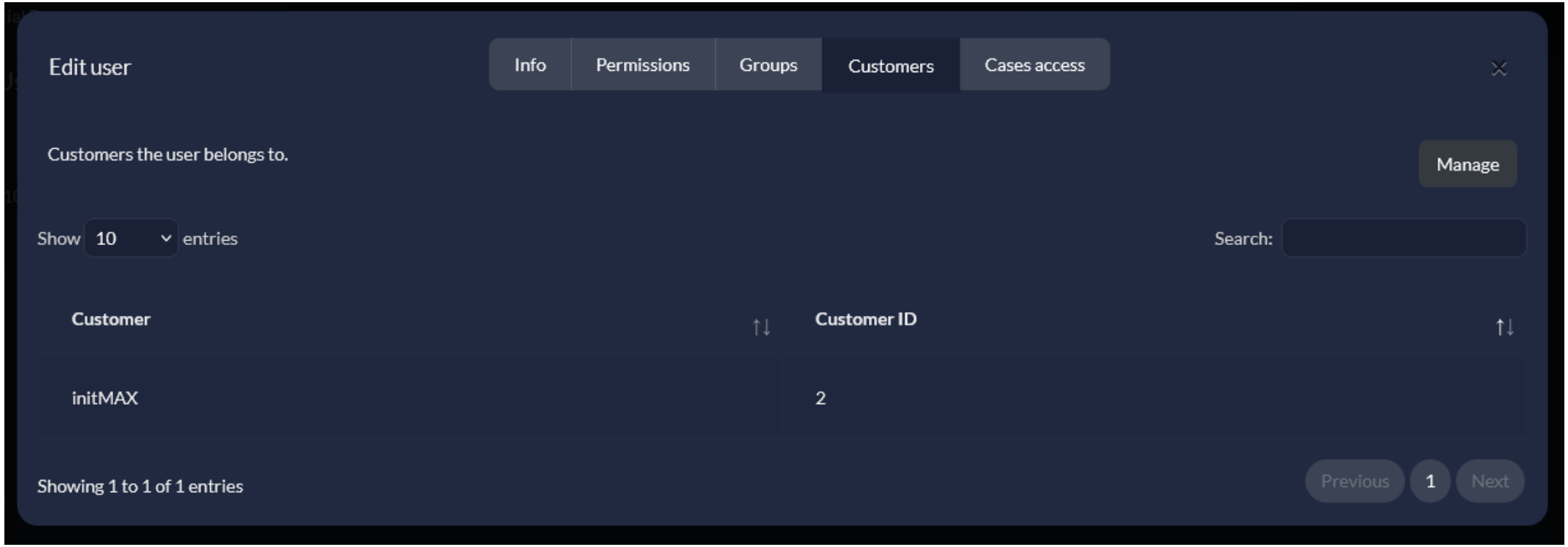
Add the user to created group:



initMAX

DFIR-IRIS setup

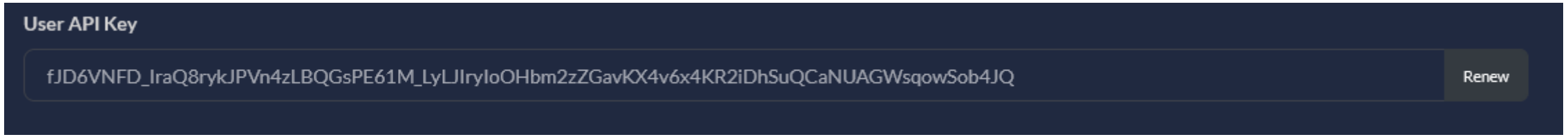
And assign the created customer to this user:



initMAX

DFIR-IRIS setup

Save the API key that has been generated for this user:



initMAX

DFIR-IRIS integration

Now it’s time to move to Wazuh integration.

Login to Wazuh Manager server and install required packages:

```
dnf install python-requests
```

Copy or create the integration script on the Wazuh Manager server in the following path:

```
/var/ossec/integrations/custom-iris.py
```

Set the correct permissions:

```
chmod +x /var/ossec/integrations/custom-iris.py
chown root.wazuh /var/ossec/integrations/custom-iris.py
```

initMAX

DFIR-IRIS integration

Copy the wrapper shell script wirth privileges:

```
cd /var/ossec/integrations/
cp –avi multiverse custom-iris
```

Modify Wazuh Manager configuration file in /var/ossec/etc/ossec.conf like this:

```
<integration>
   <name>custom-iris</name>
   <hook_url>https://<%DFIR-IRIS_IP_ADDRESS%>/alerts/add</hook_url>
   <level>7</level>
   <api_key><%DFIR-IRIS_API_KEY%></api_key>
   <alert_format>json</alert_format>
</integration>
```

initMAX

DFIR-IRIS integration

Modify the script to reflect your setup of Customer ID and your Wazuh Dashboard URL:

```
# Configuration
LOG_FILE = '/var/ossec/logs/integrations.log'
CUSTOMER_ID = 2
DASHBOARD_URL = 'https://<%WAZUH_DASHBOARD_IP_ADDRESS%>/app/wz-home'
```

Restart Wazuh Manager to reflect the changes:

```
systemctl restart wazuh-manager
```

initMAX

DFIR-IRIS integration

Check that alerts are being sent to DFIR-IRIS.

On Wazuh Manager:

```
tail -f /var/ossec/logs/integrations.log

2025-04-15 14:49:59 INFO: Alert sent to IRIS successfully. Status code: 200
```

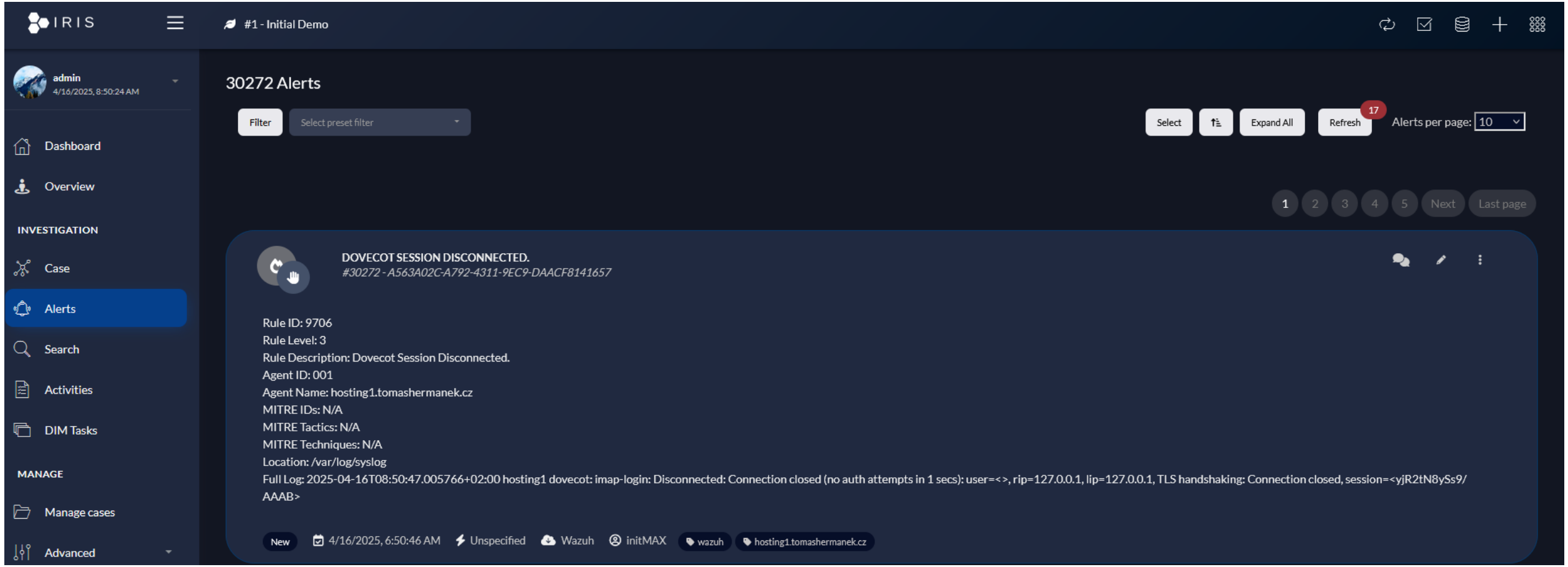
You can also check logs of Wazuh Manager itself for any errors:

```
tail -f /var/ossec/logs/ossec.log
```

Incident Response with Wazuh a DFIR-IRIS

DFIR-IRIS integration

In WebUI of DFIR-IRIS:



initMAX

Questions?

initMAX

Contact us:

Phone:	+420 800 244 442
Web:	https://www.initmax.cz
Email:	tomas.hermanek@initmax.cz
LinkedIn:	https://www.linkedin.com/company/initmax
Twitter:	https://twitter.com/initmax
Tomáš Heřmánek:	+420 732 447 184