Device Integration Guide

Overview

The SOaC Framework supports integration with various security devices and platforms to centralize detection rule management and security operations. This guide covers how to configure and manage device integrations.

Supported Device Types

1. Palo Alto Networks NGFW

Integrates with Palo Alto Networks Next-Generation Firewalls using the PAN-OS REST API.

Capabilities:
- Test connectivity
- Fetch security rules
- Retrieve threat logs
- Monitor device health

Required Configuration:
- **API URL**: Base URL of your firewall (e.g., `https://firewall.example.com`)
- **API Key**: Valid API key with read permissions

Obtaining API Key:
1. Log into Palo Alto firewall web interface
2. Navigate to **Administrator** > **Users**
3. Select your admin user
4. Generate an API key
5. Copy the key for configuration

Example Configuration:

```
{
  "api_url": "https://firewall.example.com",
  "api_key": "LUFRPT1234567890abcdefg==",
  "verify_ssl": true
}
```

Required Permissions:
- Configuration: Read
- Operational Commands: Show System Info
- Log Queries: Read threat logs

---

2. Microsoft Entra ID (Azure AD)

Integrates with Microsoft Entra ID (formerly Azure Active Directory) using Microsoft Graph API.

Capabilities:
- Test connectivity and authentication
- Fetch sign-in logs
- Query user information
- Retrieve conditional access policies
- Monitor authentication events

Required Configuration:
- **Tenant ID**: Your Azure AD tenant ID
- **Client ID**: Application (client) ID
- **Client Secret**: Client secret value

Setting Up App Registration:

1. **Register Application:**
   - Go to Azure Portal (https://portal.azure.com)
   - Navigate to **Azure Active Directory** > **App registrations**
   - Click **New registration**
   - Name: “SOaC Framework Integration”
   - Supported account types: Single tenant
   - Click **Register**

2. **Configure API Permissions:**
   - In your app registration, go to **API permissions**
   - Click **Add a permission** > **Microsoft Graph** > **Application permissions**
   - Add these permissions:
     - `AuditLog.Read.All` - Read audit log data
     - `Directory.Read.All` - Read directory data
     - `Policy.Read.All` - Read conditional access policies
     - Click **Grant admin consent**

3. **Create Client Secret:**
   - Go to **Certificates & secrets**
   - Click **New client secret**
   - Add description: “SOaC Integration”
   - Set expiration (recommend 12-24 months)
   - Click **Add**
   - **Copy the secret value immediately** (won’t be shown again)

4. **Get Tenant and Client IDs:**
   - From **Overview** page, copy:
     - Application (client) ID
     - Directory (tenant) ID

Example Configuration:

```
{
  "tenant_id": "12345678-1234-1234-1234-123456789abc",
  "client_id": "87654321-4321-4321-4321-987654321xyz",
  "client_secret": "your-client-secret-value",
  "graph_api_url": "https://graph.microsoft.com/v1.0"
}
```

Troubleshooting:
- **Error: Insufficient privileges**: Ensure admin consent is granted
- **Error: Authentication failed**: Verify tenant ID, client ID, and secret
- **Error: 403 Forbidden**: Check API permissions are added and consented

---

3. SIEM (Splunk / Elasticsearch)

Generic SIEM integration supporting Splunk and Elasticsearch platforms.

Capabilities:
- Test connectivity
- Search security events
- Query indexed logs
- Monitor platform health

Splunk Configuration

Required Configuration:
- **API URL**: Splunk REST API endpoint (e.g., `https://splunk.example.com:8089`)
- **Username**: Splunk username with search permissions
- **Password**: User password
- **SIEM Type**: `splunk`

Example Configuration:

```
{
  "api_url": "https://splunk.example.com:8089",
  "username": "admin",
  "password": "your-password",
  "siem_type": "splunk",
  "verify_ssl": true
}
```

Required Splunk Roles:
- Search permissions
- Access to relevant indexes
- REST API access enabled

Elasticsearch Configuration

Required Configuration:
- **API URL**: Elasticsearch REST API endpoint (e.g., `https://elastic.example.com:9200`)
- **Username**: Username with read permissions
- **Password**: User password
- **SIEM Type**: `elastic`

Example Configuration:

```
{
  "api_url": "https://elastic.example.com:9200",
  "username": "elastic_user",
  "password": "your-password",
  "siem_type": "elastic",
  "verify_ssl": true
}
```

Required Permissions:
- Read access to security indices
- View index metadata
- Execute search queries

Adding a Device via UI

1. **Navigate to Devices Page:**
   - Log into SOaC Framework
   - Click **Devices** in the sidebar

2. **Add New Device:**
   - Click **Add Device** button
   - Fill in the form:
     - **Device Name**: Friendly name for the device
     - **Device Type**: Select device type (Palo Alto, Entra ID, or SIEM)
     - **Configuration**: Enter device-specific credentials
     - **Enabled**: Toggle to enable/disable the device

3. **Test Connection:**
   - After creating the device, click the **Test Connection** (🔌) button
   - Verify successful connection
   - Check for any error messages

4. **Sync Rules:**
   - Once connected, click **Sync Now** (🔄) button
   - This fetches rules/configurations from the device
   - View sync results in the popup

Adding a Device via API

Create Device

```
curl -X POST http://localhost:8000/api/v1/devices \
  -H "Authorization: Bearer YOUR_TOKEN" \
  -H "Content-Type: application/json" \
  -d '{
    "name": "Production Firewall",
    "type": "paloalto",
    "enabled": true,
    "config": {
      "api_url": "https://firewall.example.com",
      "api_key": "YOUR_API_KEY"
    }
  }'
```

Test Connection

```
curl -X POST http://localhost:8000/api/v1/devices/{device_id}/test \
  -H "Authorization: Bearer YOUR_TOKEN"
```

Sync Device

```
curl -X POST http://localhost:8000/api/v1/devices/{device_id}/sync \
  -H "Authorization: Bearer YOUR_TOKEN"
```

Get Device Health

```
curl -X GET http://localhost:8000/api/v1/devices/{device_id}/health \
  -H "Authorization: Bearer YOUR_TOKEN"
```

Device Health Dashboard

Access the **Device Health** page to monitor all connected devices:

Features:
- Real-time connection status (Connected, Error, Disconnected)
- Health metrics summary
- Last tested and last sync timestamps
- Bulk refresh capability
- Quick connection testing

Status Indicators:
- 🟢 **Connected**: Device is online and responding
- 🔴 **Error**: Connection failed, check credentials/network
- ⚪ **Disconnected**: Device not tested yet

Troubleshooting

Common Issues

Connection Timeout

- **Cause**: Network connectivity, firewall rules, or incorrect URL
- **Solution**:
- Verify the device is accessible from the SOaC server
- Check firewall rules allow outbound connections
- Ensure API URL is correct and includes protocol (https://)

Authentication Failed

- **Cause**: Invalid credentials or expired tokens
- **Solution**:
- Verify API key/credentials are correct
- For Entra ID, regenerate client secret if expired
- Check user has required permissions

Permission Denied / 403 Errors

- **Cause**: Insufficient permissions on the device/platform
- **Solution**:
- Review required permissions for each device type
- Grant additional permissions as needed
- For Entra ID, ensure admin consent is granted

SSL/TLS Verification Errors

- **Cause**: Self-signed certificates or certificate mismatch
- **Solution**:
- Set `verify_ssl: false` in config (not recommended for production)
- Add proper SSL certificates to the device
- Use valid SSL certificates signed by trusted CA

Debug Mode

Enable detailed logging for troubleshooting:

```
# Backend logs
docker-compose logs -f backend

# Look for device integration messages
grep "paloalto\|entraid\|siem" backend.log
```

Security Best Practices

1. **Credential Management:**
   - Store credentials in environment variables or secret managers
   - Never commit credentials to version control
   - Rotate API keys and secrets regularly

2. **Network Security:**
   - Use HTTPS for all API connections
   - Enable SSL/TLS verification in production
   - Restrict API access to specific IP ranges

3. **Access Control:**
   - Use service accounts with minimal required permissions
   - Enable audit logging on integrated devices
   - Regularly review access logs

4. **Monitoring:**
   - Set up alerts for connection failures
   - Monitor sync success rates
   - Track authentication failures

API Endpoints Reference

Device Management

Endpoint	Method	Description
`/api/v1/devices`	GET	List all devices
`/api/v1/devices`	POST	Create new device
`/api/v1/devices/{id}`	GET	Get device details
`/api/v1/devices/{id}`	PUT	Update device
`/api/v1/devices/{id}`	DELETE	Delete device
`/api/v1/devices/{id}/test`	POST	Test connection
`/api/v1/devices/{id}/sync`	POST	Sync rules/config
`/api/v1/devices/{id}/health`	GET	Get health metrics

Environment Variables

Configure default device credentials in `.env` file:

```
# Palo Alto NGFW
PALOALTO_API_URL=https://firewall.example.com
PALOALTO_API_KEY=your-api-key
PALOALTO_VERIFY_SSL=true

# Microsoft Entra ID
ENTRAID_TENANT_ID=your-tenant-id
ENTRAID_CLIENT_ID=your-client-id
ENTRAID_CLIENT_SECRET=your-client-secret

# SIEM
SIEM_TYPE=splunk
SIEM_API_URL=https://splunk.example.com:8089
SIEM_USERNAME=admin
SIEM_PASSWORD=your-password
```

Support

For issues or questions:
- Check logs for detailed error messages
- Review device-specific documentation
- Open an issue on GitHub
- Contact your administrator

Next Steps

- Rule Management (./RULE_MANAGEMENT.md)
- API Documentation (./API.md)
- Deployment Guide (./DEPLOYMENT.md)