Micro Focus Security ArcSight SOAR 3.1

Software Version: 3.1

Integration Guides

Document Release Date: May 2021
Software Release Date: May 2021

Legal Notices

Micro Focus
The Lawn
22-30 Old Bath Road
Newbury, Berkshire RG14 1QN
UK

https://www.microfocus.com

Copyright Notice

Trademark Notices

Documentation Updates

The title page of this document contains the following identifying information:

- Software Version number
- Document Release Date, which changes each time the document is updated
- Software Release Date, which indicates the release date of this version of the software

To check for recent updates or to verify that you are using the most recent edition of a document, go to:

https://www.microfocus.com/support-and-services/documentation

Contents

Integration Guide for AlientVault OTX 17
- Integration Overview 17
- Integration Capabilities 17
- Configuration 17
  - Configuring SOAR 18
  - Additional Notes 20

Integration Guide for Amazon EC2 22
- Integration Overview 22
- Integration Capabilities 22
- Configuration 22
  - Prerequisites 22
  - Configuring on Amazon AWS 23
  - Configuring on SOAR 27
  - Additional Notes 29

Integration Guide for Anomali ThreatStream 30
- Integration Overview 30
- Integration Capabilities 30
  - Configuration 31
    - Prerequisites 31
    - Configuring Anomali ThreatStream 31
    - Configuring SOAR 31
  - Additoinal Notes 34

Integration Guide for Arbor Networks APS 35
- Integration Overview 35
- Integration Capabilities 35
- Configuration 35
  - Prerequisites 35
  - Configuring Arbor Networks APS 35
  - Configuring SOAR 36

Integration Guide for Bind RPZ DNS 38
- Integration Overview 38
- Integration Capabilities 38
- Configuration 38

Prerequisites 38
Configuring SOAR 38
Integration Guide for Carbon Black Response (EDR) 41
Integration Overview 41
Integration Capabilities 41
Prerequisites 42
Configuring Carbon Black Response(EDR) 42
Configuring SOAR 42
Additional Notes 44
Integration Guide for Check Point R80 45
Integration Overview 45
Integration Capabilities 45
Configuration 45
Configuring Check Point R80 45
Configuring Access Rights 46
Configuring SOAR 47
Integration Guide for Check Point SandBlast 50
Integration Overview 50
Integration Capabilities 50
Configuration 50
Prerequisites 50
Configuring Check Point SandBlast 50
Configuring SOAR 51
Integration Guide for Cisco ASA Firewall 54
Integration Overview 54
Integration Capabilities 54
Configuration 54
Prerequisites 54
Configuring Cisco ASA Firewall 54
Configuring SOAR 55
Integration Guide for Cisco Firepower Management Center 57
Integration Overview 57
Integration Capabilities 57
Configuration 57
Prerequisites 57
Configuring Cisco Firepower Management Center 58

Configuring SOAR 60
Integration Guide for Cisco Identity Service Engine 63
Integration Overview 63
Integration Capabilities 63
Configuration 63
Prerequisites 63
Configuring Cisco Identity Services Engine 63
Configuring SOAR 63
Integration Guide for Cisco Ironport Email Security 65
Integration Overview 65
Integration Capabilities 65
Configuration 65
Prerequisites 65
Configuring Cisco Ironport Email Security 65
Configuring SOAR 66
Additional Notes 67
Integration Guide for Cyberark Central Credential Provider 68
Integration Overview 68
Configuration 68
Prerequisites 68
Configuring CyberArk Application Identity Manager 68
Configuring SOAR 69
Additional Notes 70
Integration Guide for CYMRU Malware Hash Registry Query 71
Integration Overview 71
Integration Capabilities 71
Configuration 71
Configuring CYMRU Malware Hash Registry Query 71
Configuring SOAR 71
Integration Guide for CyThreat Threat Intelligence 73
Integration Overview 73
Integration Capabilities 73
Configuration 73
Prerequisites 73
Configuring CyThreat Threat Intelligence 73
Configuring SOAR 74

Configuring Credentials 74
Configuring CyThreat Threat Intelligence Feed as Alert Source 74
Additional Notes 78
Integration Guide for DNS Service 79
Integration Overview 79
Integration Capabilities 79
Configuration 79
Prerequisites 79
Configuring DNS Service 79
Configuring SOAR 79
Integration Guide for ESB Karar 81
Integration Guide for F5 Big-IP Advanced Firewall Manager 83
Integration Overview 83
Integration Capabilities 83
Configuration 83
Configuring F5 Big-IP Advanced Firewall Manager 83
Integration Guide for FireEye HX 85
Integration Overview 85
Integration Capabilities 85
Enrichment 85
Action 85
Configuring FireEye HX 85
Configuring SOAR 85
Integration Guide for Forcepoint Cloud Services 88
Integration Overview 88
Integration Capabilities 88
Configuration 88
Configuring Forcepoint Cloud Services 88
Configuring SOAR 88
Integration Guide for Forcepoint Content Gateway 90
Integration Overview 90
Integration Capabilities 90
Configuration 90
Prerequisites 90
Configuring Facepoint Web Content Gateway 91
Configuring SOAR 91

Integration Guide for ForeScout CounterACT NAC 94
Integration Overview 94
Integration Capabilities 94
Configuration 94
Prerequisites 94
Configuring ForeScout CounterACT NAC 95
Configuring SOAR 97
Additional Notes 100
Integration Guide for Fortinet FortiGate Firewall 102
Integration Overview 102
Integration Capabilities 102
Configuration 102
Configuring FortiGate Firewall 102
Configuring SOAR 102
Additional Notes 103
Integration Guide for Fortinet FortiAnalyzer 105
Integration Overview 105
Integration Capabilities 105
Configuring Fortinet FortiAnalyzer 105
Configuring SOAR 105
Integration Guide for Fortinet FortiDDoS 108
Integration Overview 108
Integration Capabilities 108
Configuration 108
Prerequisites 108
Configuring FortiDDoS 108
Configuring SOAR 109
Integration Guide for Fortinet FortiGate API 112
Integration Overview 112
Integration Capabilities 112
Configuration 112
Prerequisites 112
Configuring Fortinet FortiGate 112
Configuring SOAR 115
Additional Notes 116
Integration Guide for Fortinet FortiMail 117

Integration Overview 117
Integration Capabilities 117
Configuration 117
Prerequisites 117
Configuring FortiMail 117
Configuring SOAR 119
Additional Notes 120
Integration Guide for Fortinet FortiManager 122
Integration Overview 122
Integration Capabilities 122
Prerequisites 122
Configuring FortiManager 123
Configuring SOAR 124
Additional Notes 125
Integration Guide for Fortinet FortiSandbox 126
Integration Overview 126
Integration Capabilities 126
Configuration 126
Prerequisites 126
Configuring Fortinet Sandbox 126
Configuring SOAR 128
Integration Guide for FTP Server 131
Integration Overview 131
Integration Capabilities 131
Configuration 131
Prerequisites 131
Configuring SOAR 131
Integration Guide for Generic HTTP SMS Gateway 134
Integration Overview 134
Integration Capabilities 134
Configuration 134
Configuring Generic HTTP SMS Gateway 134
Configuring SOAR 134
Integration Guide for HTTP Proxy 136
Integration Overview 136
Configuration 136

Prerequisites 136
Configuring HTTP Proxy 136
Configuring SOAR 136
Integration Guide for IBM Security QRadar 139
Integration Overview 139
Integration Capabilities 139
Configuration 140
Prerequisites 140
Configuring IBM QRadar 140
Configuring SOAR 145
Configuring IBM QRadar as Alert Source 146
Configuring IBM QRadar as Integration 150
Integration Guide for IBM Security X-Force 154
Integration Overview 154
Integration Capabilities 154
Configuration 154
Prerequisites 154
Configuring IBM X-Force Exchange 155
Configuring SOAR 155
Integration Guide for Infoblox DNS Firewall 158
Integration Overview 158
Integration Capabilities 158
Configuration 158
Prerequisites 158
Configuring Infoblox DNS Firewall 158
Configuring SOAR 159
Integration Guide for Invictus USTA ThreatIntelligence 162
Integration Overview 162
Integration Capabilities 162
Configuration 162
Prerequisites 162
Configuring Invictus USPA 163
Configuring SOAR 163
163
Configuring Invictus USTA as Alert Source 163
Configuring Invictus USTA as Integration 164
Additional Notes 165

Integration Guide for JDBC(Database) Server 166
Integration Capabilities 166
Configuration 166
Prerequisites 166
Configuring Database Server 166
Configuring SOAR 166
Integration Guide for Juniper SRX Firewall 169
Integration Overview 169
Integration Capabilities 169
Configuration 169
Configuring Juniper SRX Firewall 169
Configuring SOAR 169
Integration Guides for Kannel SMS Gateway 171
Integration Overview 171
Integration Capabilities 171
Supported Action Capabilities 171
Configuring Kannel SMS Gateway 171
Configuring SOAR 171
Integration Guide for Karmasis Infraskope 174
Integration Overview 174
Integration Capabilities 174
Configuration 174
Prerequisites 174
Configuring Karmasis Infraskope 174
Configuring SOAR 174
Configuring Karmasis Infraskope as Alert Source 175
Integration Guide for Kaspersky Security Center 179
Configuration on Kaspersky Security Center 179
Configuring SOAR 179
Optional configuration 180
Overriding built-in scripts 180
Get Task Names 181
Get Group Names 182
Get Tag Names 183
Host Information Enrichment 184
Block Hash Action Capability 187
Rollback of block hash capability 189

Add tag to host capability 191
Rollback of Add Tag to Host Capability 192
Move system to group capability 194
Run task capability 195
Integration Guide for LogRhythm SIEM 197
Prerequisites 197
Configuration on LogRhythm SIEM 198
Configuration on ATAR 201
Configuring LogRhythm SIEM as Alert Source 201
Configuring LogRhythm SIEM as Integration 203
Integration Guide for MAY Siber Scop NET 204
Prerequisites 204
Configuring MAY Siber Scop NET 204
Configuring SOAR 204
Integration Guide for MAY Siber SIEM 207
Configuration 207
Integration Guide for McAfee Enterprise Security Manager (ESM) 214
Prerequisites 214
Configuration on McAfee Enterprise Security Manager (ESM) 215
Configuring SOAR 215
Integration Guide for McAfee ePolicy Orchestrator 217
Prerequisites 217
Configuration on McAfee ePolicy Orchestrator 218
Configuring SOAR 218
Integration Guide for McAfee Network Security Platform (IPS) 220
Configuration 220
Configuration on McAfee Network Security Platform (IPS) 220
Configuring SOAR 220
Integration Guide for McAfee Web Gateway 222
Prerequisites 222
Configuration on McAfee Web Gateway 222
Configuration on SOAR 223
Integration Guide for Micro Focus Arcsight ESM 225
Prerequisites 225
Configuration on Micro Focus Arcsight ESM 226
Configuration on SOAR 230

Configuring Credentials 230
Configuring Micro Focus Arcsight ESM as Alert Source 231
Configuring Micro Focus Arcsight ESM as Integration 232
Additional Notes 233
Integration Guide for Micro Focus ArcSight Intelligence 235
Configuring ArcSight Intelligence 236
Configuring SOAR 236
Configuring Micro Focus ArcSight Intelligence as Alert Source 236
Configuring Micro Focus ArcSight Intelligence as Integration 238
Integration Guide for Micro Focus Arcsight Logger 241
Configuration 241
Prereqisites 241
Configuration on Micro Focus Arcsight Logger 241
Configuring SOAR 241
Configuring SOAR 241
Integration Guide for Microsoft Active Directory 244
Configuration 244
Prerequisites 244
Configuration on Microsoft Active Directory 245
Configuring SOAR 245
Integration Guide for Microsoft Exchange 248
Prerequisites 248
Configuration on Microsoft Exchange 248
Additional Notes 250
Integration Guide for Microsoft Office365 Exchange EWS 251
Configuration on Microsoft Exchange 252
Configuring SOAR 252
Additional Notes 254
Integration Guide for Microsoft Windows DNS Server 255
Configuration on Microsoft Windows DNS Server 255
Configuring ATAR 255
Integration Guide for Microsoft Windows Services (WinRM) 257
Configuration on Microsoft Windows Services 257
Configuring SOAR 257
Integration Guide for MISP 259
Integration Overview 259

Integration Capabilities 259
Prerequisites 259
Integration Guide for Ones BioAffix 262
Integration Capabilities 262
Prerequisites 262
Configuration on Ones BioAffix 262
Configuring SOAR 262
Additional Notes 263
Integration Guide for Palo Alto Networks AutoFocus 264
Prerequisites 264
Configuration on Palo Alto Networks AutoFocus 265
Configuring SOAR 265
Integration Guide for Palo Alto Networks Firewall 267
Prerequisites 267
Configuration on Palo Alto Networks Firewall (API) 267
Configuring SOAR 268
Additional Notes 269
Integration Guide for Palo Alto Networks Panorama 270
Prerequisites 270
Configuration on Palo Alto Networks Panorama 271
Configuration on SOAR 271
Integration Guide for Recorded Future 273
Prerequisites 273
Configuration on Recorded Future 274
Configuring SOAR 274
Integration Guide for Robtex Lookup 276
Configuration on Robtex Lookup 276
Configuring SOAR 276
Integration Guide for Roksit DNS Firewall 278
Prerequisites 278
Configuration on Roksit DNS Firewall 278
Configuring SOAR 278
Integration Guide for RSA Envision Configuration 280
Integration Guide for RSA Security Analytics 281
Prerequisites 281

Configuration on RSA Security Analytics Suite 282
Configuring SOAR 282
Integration Guide for SMTP Mail Server 284
Prerequisites 284
Configuring SOAR 284
Additional Notes 285
Integration Guide for Sophos XG Firewall 286
Prerequisites 286
Configuration on Sophos XG Firewall 286
Integration Guide for SORBS Query 289
Configuration on SORBS Query 289
Configuring SOAR 289
Integration Guide for Splunk Enterprise Security (Alert Source) 291
Configuring Splunk Enterprise Security 291
Configuring SOAR 291
SOAR Configuration Parameters 294
Additional Notes 295
Integration Guide for Splunk Enterprise Security 296
Configuring SOAR 296
Integration Guide for Symantec Advanced Threat Protection 297
Configuring Symantec Advanced Threat Protection 297
297
Configuring SOAR 297
Integration Guide for Symantec Bluecoat Malware Analysis Appliance (MAA) 299
Prerequisites 299
Configuring Symantec Bluecoat Malware Analysis Appliance (MAA) 299
Configuring SOAR 299
Integration Guide for Symantec BlueCoat Proxy SG 301
Prerequisites 301
Configuring Symantec BlueCoat Proxy SG 302
Configuring SOAR 302
Integration Guide for Symantec Bluecoat Site Review 305
Configuration on Bluecoat Site Review 305
Configuring SOAR 305
Integration Guide for Symantec Data Loss Prevention (DLP) 306

Integration Capabilities 306
Prerequisites 306
Configuring Symantec DLP 306
Configuring SOAR 307
Integration Guide for Symantec DeepSight Intelligence 309
Prerequisites 309
Configuring Symantec DeepSight Intelligence 310
Configuring SOAR 310
Configuring Symantec DeepSight Intelligence as Alert Source 311
Configuring Symentec DeepSight Intelligence as Integration 312
Integration Guide for Symantec Endpoint Protection Manager 313
Prerequisites 313
Configuring Symantec Endpoint Protection Manager 314
Configuring SOAR 314
Integration Guide for Symantec Managed Security Services (MSS) 317
Configuring Symantec MSS 318
Configuring SOAR 318
Configuring Credentials 318
Configuring Symantec MSS as Alert Source 319
Configuring Symantec MSS as an Integration 321
Additional Notes 322
Integration Guide for Symantec Messaging Gateway 324
Prerequisites 324
Configuring Symantec Messaging Gateway 324
Configuring SOAR 325
Integration Guide for Tenable Nessus 327
ConfiguringTenable Nessus 327
Configuration on SOAR 327
Configuring SOAR 327
Integration Guide for Tenable Security Center 329
Prerequisites 329
Configuring Tenable Security Center 329
Configuring SOAR 330
Integration Guide for Trend Micro Control Manager 331
Configuring Trend Micro Control Manager 331
Configuring SOAR 331

Integration Guide for Turkcell Threat Intelligence or Bozok 333
Prerequisites 333
Configuration on Turkcell Threat Intelligence or Bozok 333
Configuring SOAR 334
Integration Guide for USOM (TR-CERT) Intelligence Feed 336
Prerequisites 336
Configuring USOM (TR-CERT) Intelligence Feed 336
Configuring SOAR 336
Configuring USOM (TR-CERT) Intelligence Feed as Alert Source 336
Additional Notes 338
Integration Guide for VirusTotal 339
Prerequisites 339
Configuring VirusTotal 340
Configuring SOAR 340
Additional Notes 343
Integration Guide for VMware ESXi 344
Configuring VMware ESXi 344
Configuring SOAR 344
Integration Guide for VxStream Sandbox 346
Configuration on VxStream Sandbox 346
Configuring SOAR 346
Integration Guide for WinRM 349
Configuring SOAR 351
Configuring Domain-Controller for WinRM HTTPS Transport 351
Force Group Policy Update 353
Additional Notes 354

Send Documentation Feedback 355

Integration Guide for AlientVault OTX

Integration Overview

AlienVault OTX is an open threat exchange platform supported by AlienVault and the community.

Integration Capabilities

ArcSight SOAR has the following integration capabilities with AlienVault OTX:

- IP Indicator
- Hash Indicator
- URL Indicator
- Domain Indicator
- Hostname Indicator

Use Case: Enrichment of artifacts detected in the organization

SOAR, when integrated with AlienValut OTX, can search for an artifact and gather information such as related threats and recent detections. This information may lead the investigation into a different path, and analysts can investigate and root out malicious activities in their networks.

This integration can be performed automatically within a playbook or manually by an analyst.

Configuration

Prerequisites

- SOAR connects to AlienVault OTX API via HTTPS. Typically it runs on 443/tcp port. So access to this service is required.
- A user account is required for SOAR to connect to AlienVault OTX. It can be created from the following link:

  https://otx.alienvault.com

Configuring AlienVault OTX

- AlienVault OTX requires an API key for access. Users can retrieve it from https://otx.alienvault.com/api after logging in with a valid credential.

Configuring SOAR

1. Click Configuration > Credentials > Create Credential
2. Fill in the Credential Editor form with the following information:

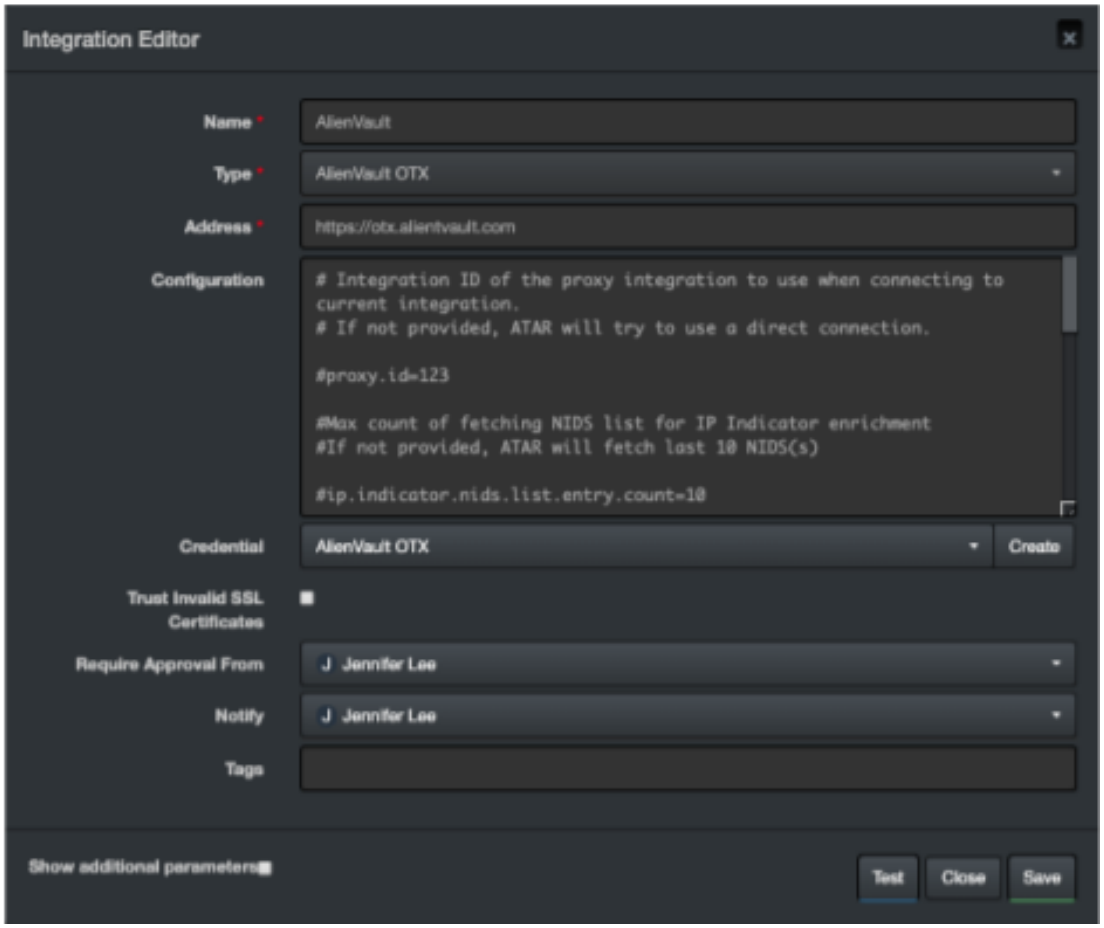
   a. **Type:** Internal Credential

   - **Name**: Display name of credential set (i.e., AlienVault OTX Credentials)
   - **Username**: Empty
   - **Password**: Empty
   - **Private Key**: API Key retrieved from the AlienVault OTX
3. Click Configuration > Integrations > Create Integration
4. Fill in the configuration form with the following information:
   - **Name:** Display name of AlienVault OTX integration on SOAR
   - **Type:** AlienVault OTX
   - **Address:** Address of the cloud service is standard: https://otx.alienvault.com
   - **Configuration:** You need to specify the following configuration parameters

```
# Integration ID of the proxy integration to use when connecting to
current

# integration.

# If not provided, SOAR will try to use a direct connection.

#proxy.id=123

#Max count of fetching NIDS list for IP Indicator enrichment

#If not provided, SOAR will fetch last 10 NIDS(s)

#ip.indicator.nids.list.entry.count=10

#Max count of fetching URL list for IP Indicator enrichment

#If not provided, SOAR will fetch last 50 URL(s)

#ip.indicator.url.list.entry.count=50
```

```
#Max count of fetching URL list for Domain Indicator enrichment

#If not provided, SOAR will fetch last 50 URL(s)

#domain.indicator.url.list.entry.count=50

#Max count of fetching Malware list for Hostname Indicator enrichment

#If not provided, SOAR will fetch last 50 Malware(s)

#hostname.indicator.malware.list.entry.count=50

#Max count of fetching URL list for Hostname Indicator enrichment

#If not provided, SOAR will fetch last 50 URL(s)

#hostname.indicator.url.list.entry.count=50

# configure how far (in minutes) into the past this enrichment will
look.

#cache.reusing.duration=20
```

- **Credential**: Name of the credential set you've just created on step 2. (i.e., AlienVault OTX Credentials).
- **Trust Invalid SSL Certificates:** Select this if Engine's certificate is self-signed or not recognized by browsers. Not selected.
- **Require Approval From:** Select user(s) from list to ask her/his approval before executing enrichments on this integration.
- **Notify**: Select user(s) from the list to notify when SOAR performs an enrichment on this integration.

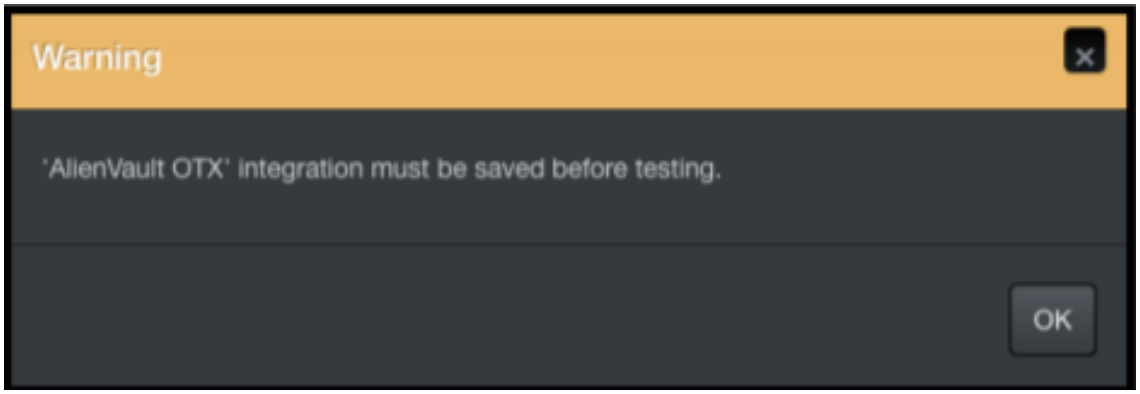


5. When you click the Test button the following pop up should be displayed if your credentials and address are valid.

6. Click Save to complete integration.

Additional Notes

- AlienVault OTX integration on SOAR is an Advanced Action Script, and the content of the default script is accessible under Configuration > Customization Library.
- While defining the integration for the first time, you will encounter the following warning message, which is expected behavior for this type of integration.


Warning
'AlienVault OTX' integration must be saved before testing.
OK

Integration Guide for Amazon EC2

Integration Overview

Amazon EC2 (Elastic Compute Cloud) forms a central part of Amazon.com's cloud-computing platform, Amazon Web Services, by allowing users to establish virtual networks and rent virtual computers on which they can run their own applications. Amazon EC2 REST-API supports the following Amazon Web Services:

- Amazon EC2
- Amazon EBS
- Amazon VPC
- AWS VPN

Please note that this integration is in Beta.

Integration Capabilities

ArcSight SOAR has the following integration capabilities with Amazon EC2:

- Add Network ACL Entry (VPC)
- Delete Network ACL Entry (VPC)

Use Case: Blocking Attackers

SOAR when integrated with Amazon EC2, blocks the attacker's IP addresses while responding to a cyber-attack. The blocking can be performed automatically within a playbook or manually by an analyst.

Configuration

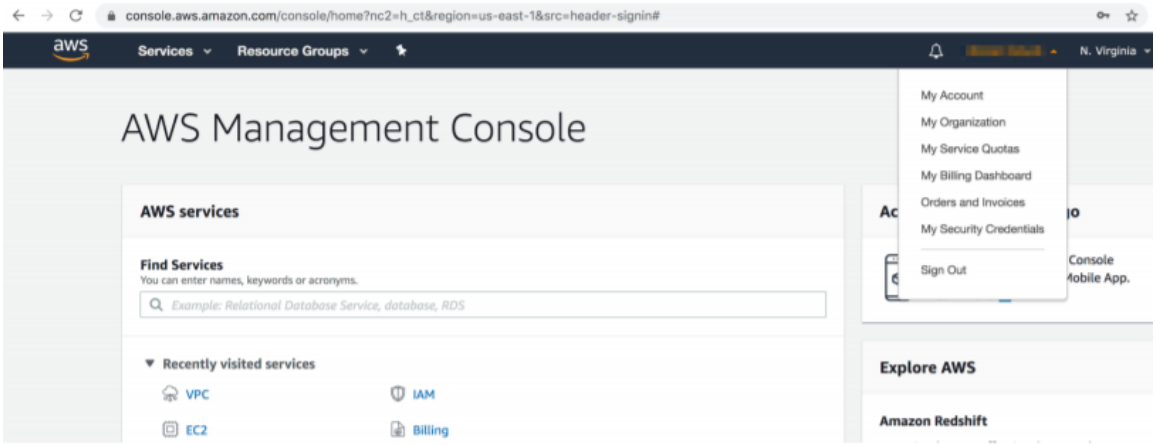
Prerequisites

- SOAR connects to Amazon EC2 API via HTTPS. Access to https://ec2.amazonaws.com (443/tcp port) is required.
- AWS Access Key and AWS Access Key Secret are required for SOAR to connect

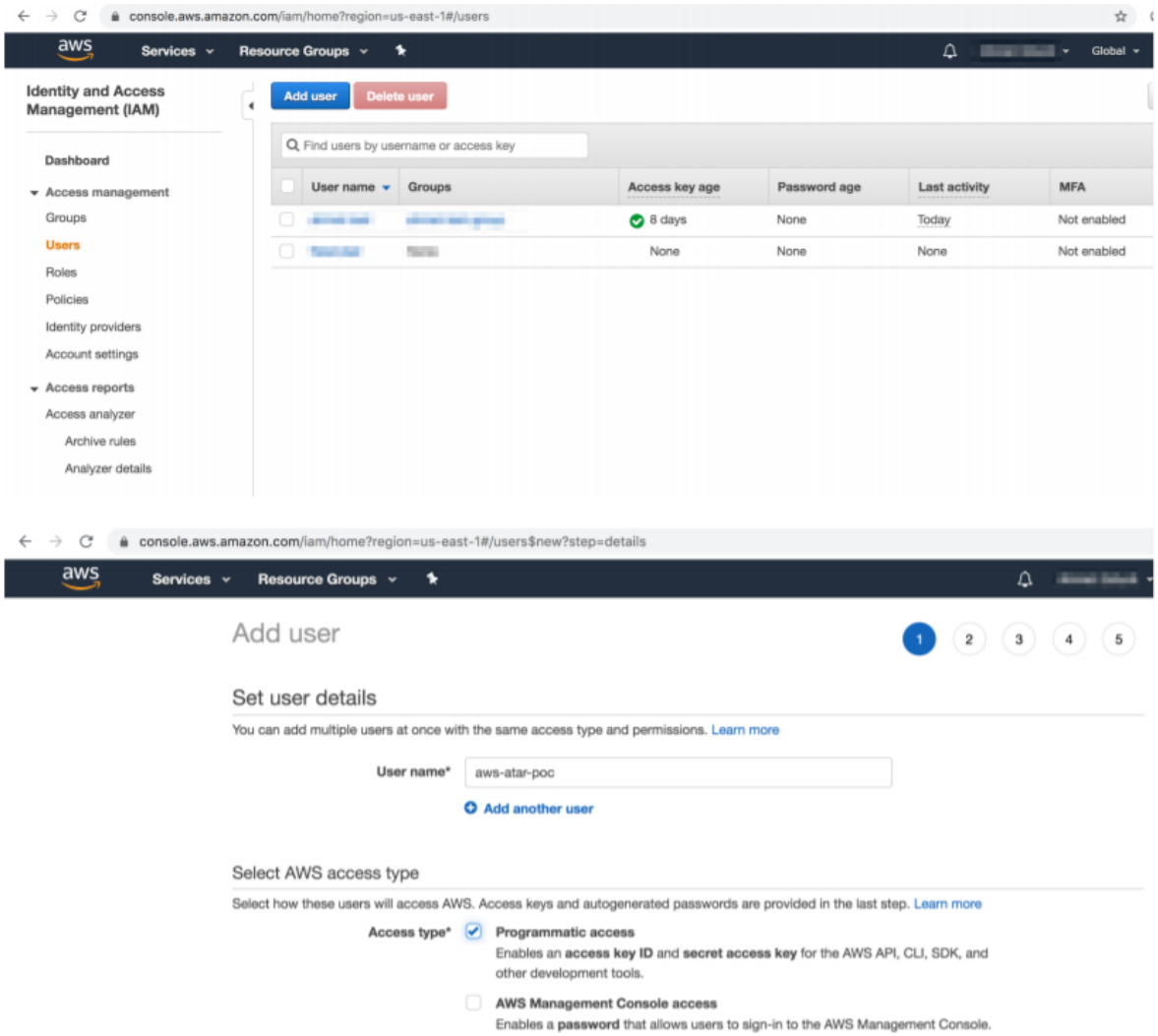
Amazon Web Services.

Configuring on Amazon AWS

1. Log in to Amazon Console (https://aws.amazon.com). Navigate to My Security Credentials, and select Identity Access Management (IAM) service:

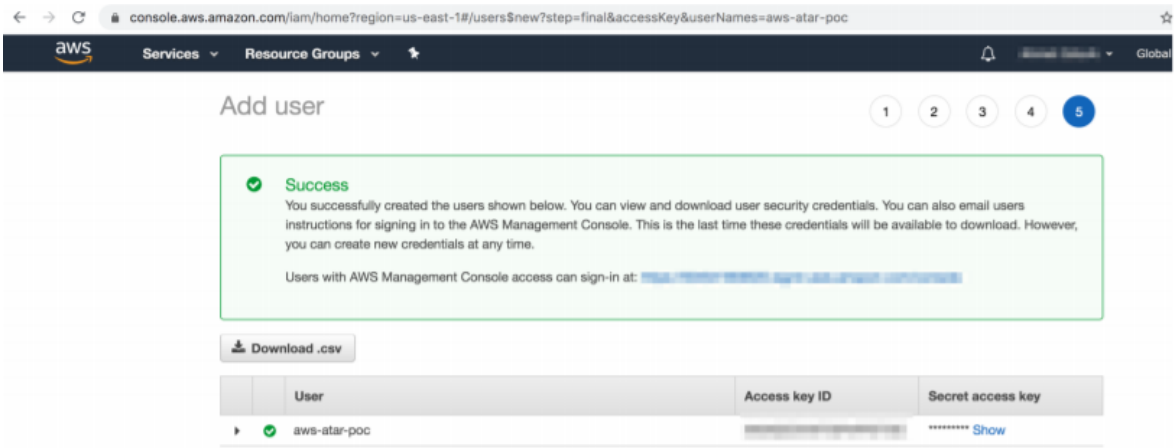


2. To add an IAM(identity and access management) user, click Access Management > Users > Add User. While adding new user account, it is important to select Access Type as Programmatic Access.



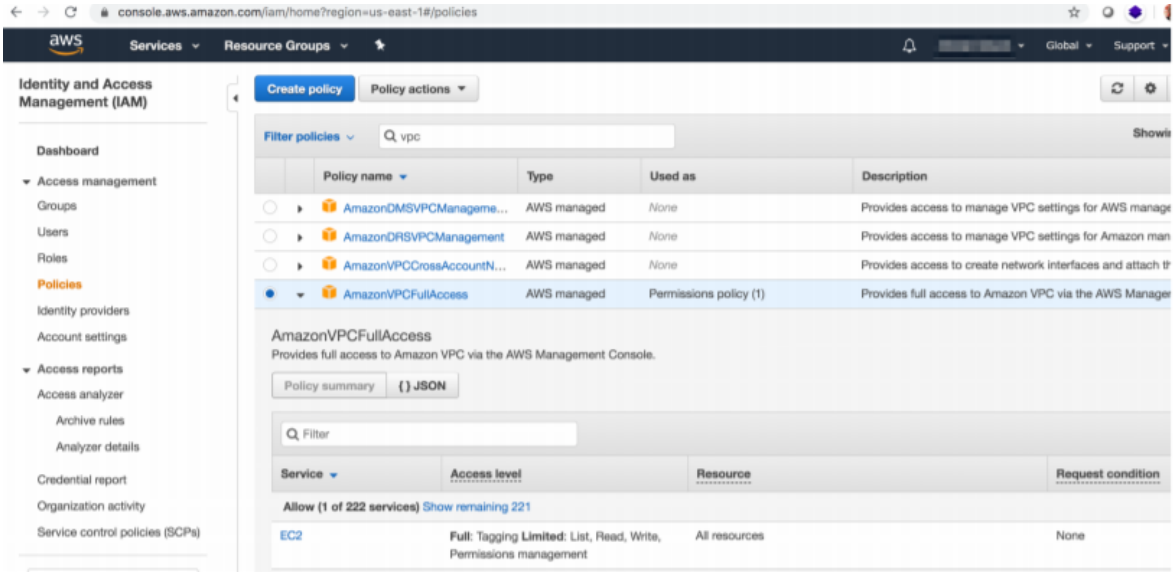
3. You can skip the next steps in the Add User process until Access Key and Access Key Secret are displayed.

Note: Download the credentials as the Access Key Secret is never displayed post this step.



4. To arrange access policy, click > Access Management > Policies, and search for the required policy in previously defined policies list.

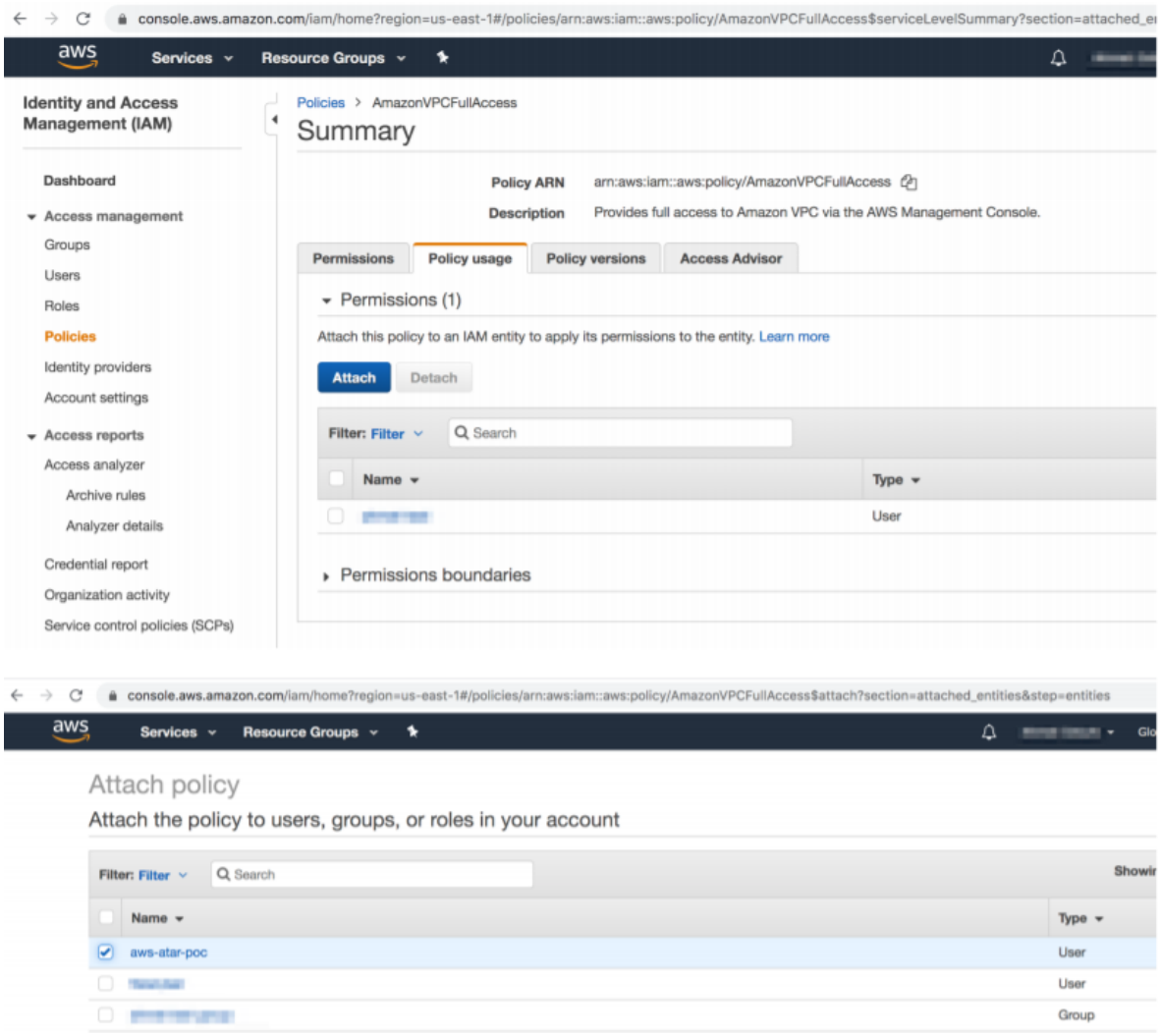
   For examle, the following image shows the policy AmazonVPCFullAccess.



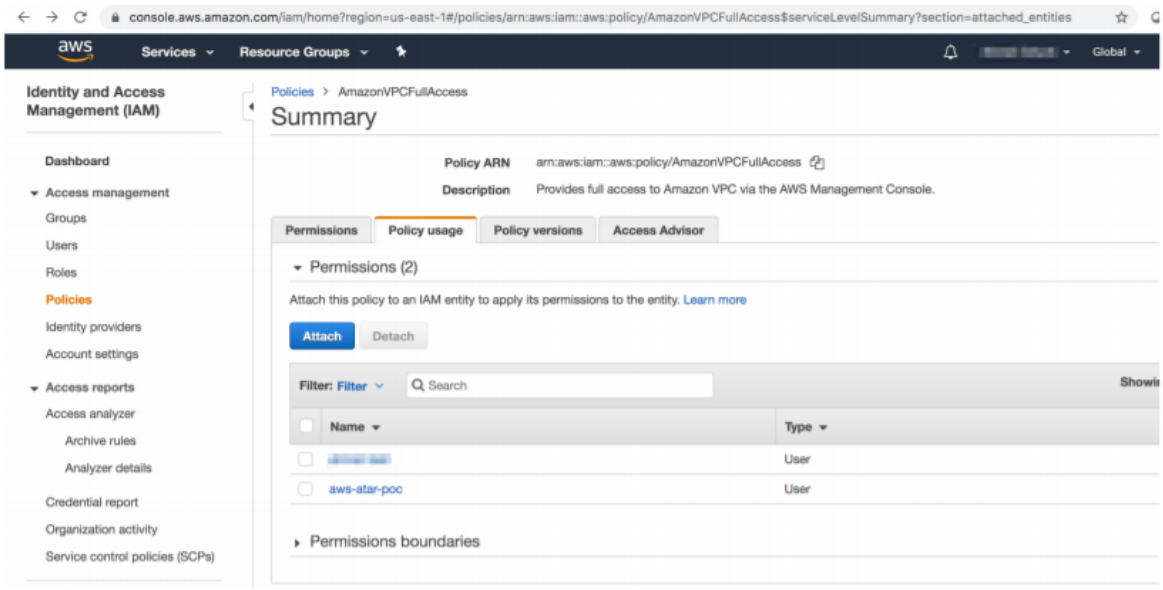
5. Select AmazonVPCFullAccess and open the Policy Summary.

   a. **Click Policy Usage** > **Attach**.

   b. In the Attach Policy menu, select the user that you have created in the previous steps, from the available users list in the system.



6. You can verify if the permission is successful for the user account that you've created on the Policy Usage page.



Configuring on SOAR

1. Click **Configuration** > **Credentials** > **Create Credential.**
2. Fill the Credential Editor form with the following information:

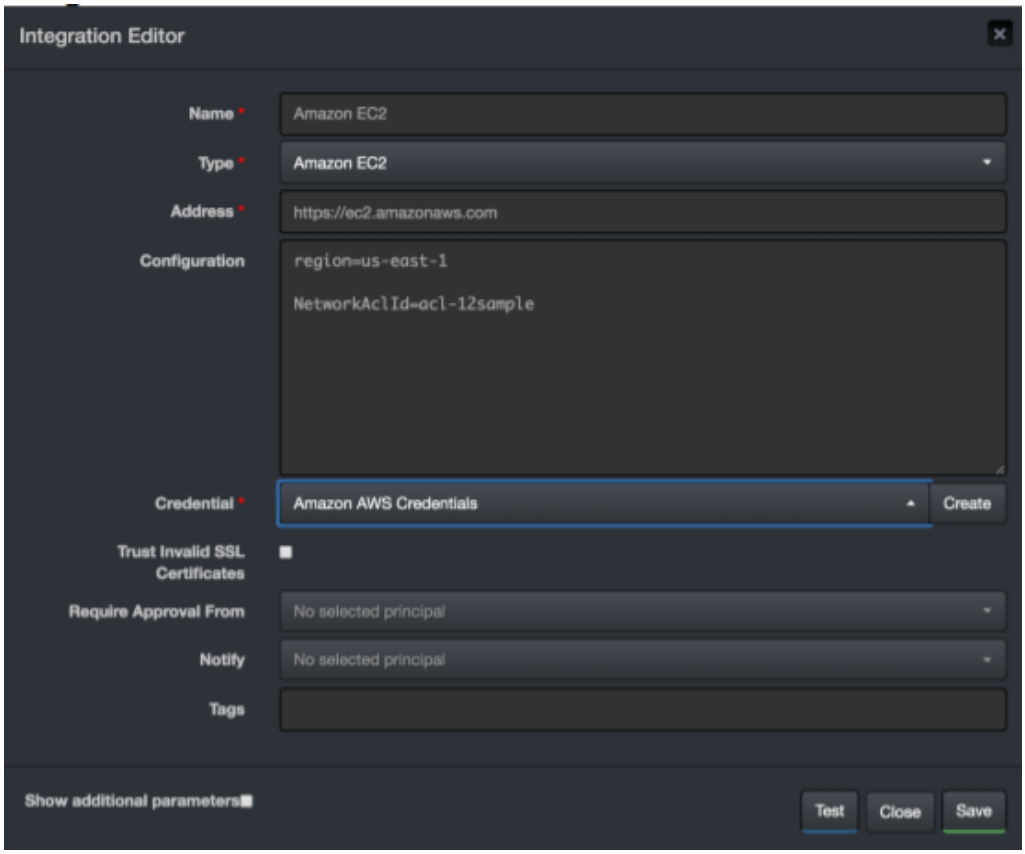
   a. Internal Credential:

   - **Type:** Internal credential
   - **Name:** Display name of credential set (i.e., Amazon AWS Credentials)
   - **Username:** Access Key of IAM user you have created
   - **Password:** Secret of Access Key of IAM user you have created
   - **Private Key:** Empty

   b. Credential Store:

   - **Type:** External credential
   - **Name:** Name of the credential with full path of the safe on store
3. Click Configuration > Integrations > Create Integration. Fill the Configuration form with the following information:
   - **Name:** Display name of Amazon EC2 integration on SOAR
   - **Type:** Amazon EC2

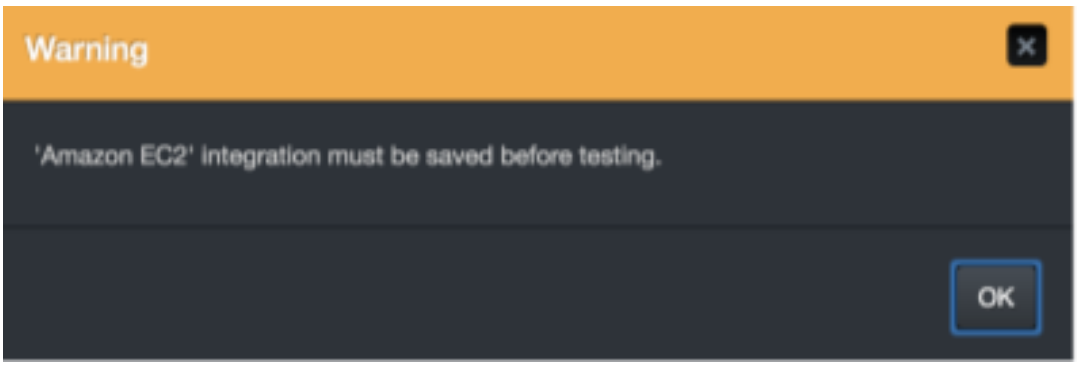
- **Address:** Address of the integration (https://ec2.amazonaws.com)
- **Configuration:** You need to specify the following configuration parameters
- **Credential:** Name of the credential set you have just created on step 2. (i.e., Amazon AWS Credentials)
- **Trust Invalid SSL Certificates:** No need to select
- **Require Approval From:** Select user(s) from list to ask her/his approval before executing actions on this integration
- **Notify:** Select user(s) from the list to notify when SOAR performs an action on integration



4. Click the Test button. The following pop up will be displayed if your credential and address are valid.
5. Click Save to complete integration.

Additional Notes

- Amazon EC2 integration on SOAR is an Advanced Script, and the content of the default script is accessible under **Configuration** > **Customization** Library.
- While defining the integration for the first time, you might encounter the following warning message, which is the expected behavior for this type of integration.



Integration Guide for Anomali ThreatStream

Integration Overview

Anomali ThreatStream is a Threat Intelligence Platform that enables businesses to integrate security products and leverage threat data to defend against cyber threats.

Integration Capabilities

ArcSight SOAR has the following integration capabilities with Anomali ThreatStream:

- Domain Reputation
- Email Reputation
- File Reputation
- Get Incident Details
- Get Intelligence
- Get Investigation Details
- IP Reputation
- List Incidents
- List Investigations
- Report Indicator
- Create Investigation
- Close Investigation
- Update Investigation

Use Case: Investigating Phishing Campaigns

SOAR, when integrated with Anomali ThreatStream, helps campaigns that investigate and mitigate phishing. When a phishing report email comes from a user, SOAR extracts the indicators such as IP address, URLs and attachments in the message and creates an incident on the Incident Management Service Desk. SOAR then checks with Anomali ThreatStream, to know if this is a known attack and whether these indicators were previously analyzed.

This investigation can be either performed automatically within a playbook or manually by an analyst.

Configuration

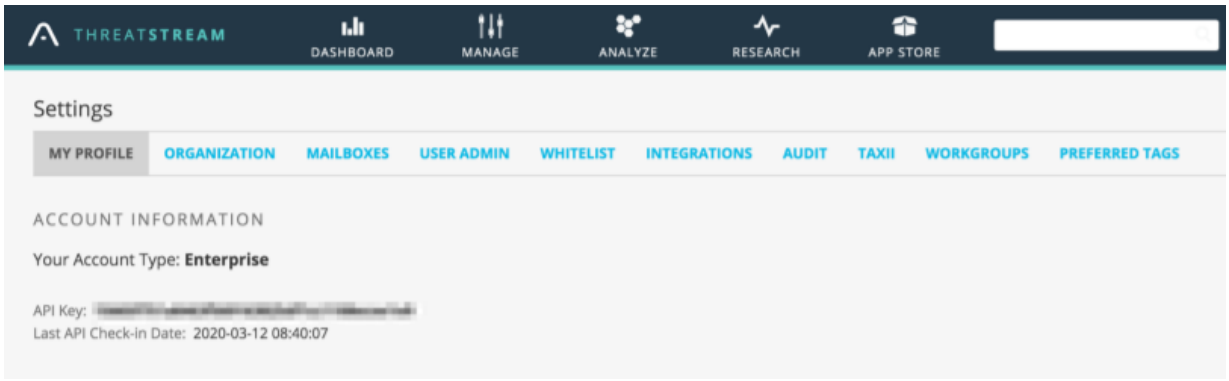
Prerequisites

- SOAR connects to Anomali ThreatStream API via HTTPS. Access to https://api.threatstream.com/ **(443/tcp port)** is required.
- An API key is required for SOAR to connect to Anomali ThreatStream Service.

Configuring Anomali ThreatStream

1. Log in to https://ui.threatstream.com/.
2. Navigate to **Settings** > **My Profile** to get the API Key.

Note: This key is required by SOAR to access the platform for queries.



Configuring SOAR

1. **Configuration** > **Credentials** > **Create Credential.**
2. Fill the **Credential Editor** form with the following details:

   a. **Internal Credential:**

Parameter	Value
Type	Internal credential
Name	Display name of credential set (For example, Anomali ThreatStream Credentials)
Username	Your username on Anomali ThreatStream platform
Password	Empty
Private Key	API key you have obtained from Anamoli ThreatStream Platform

b. **Credential Store:**

Parameter	Value
Type	External credential
Name	Name of the credential with full path of the safe on store

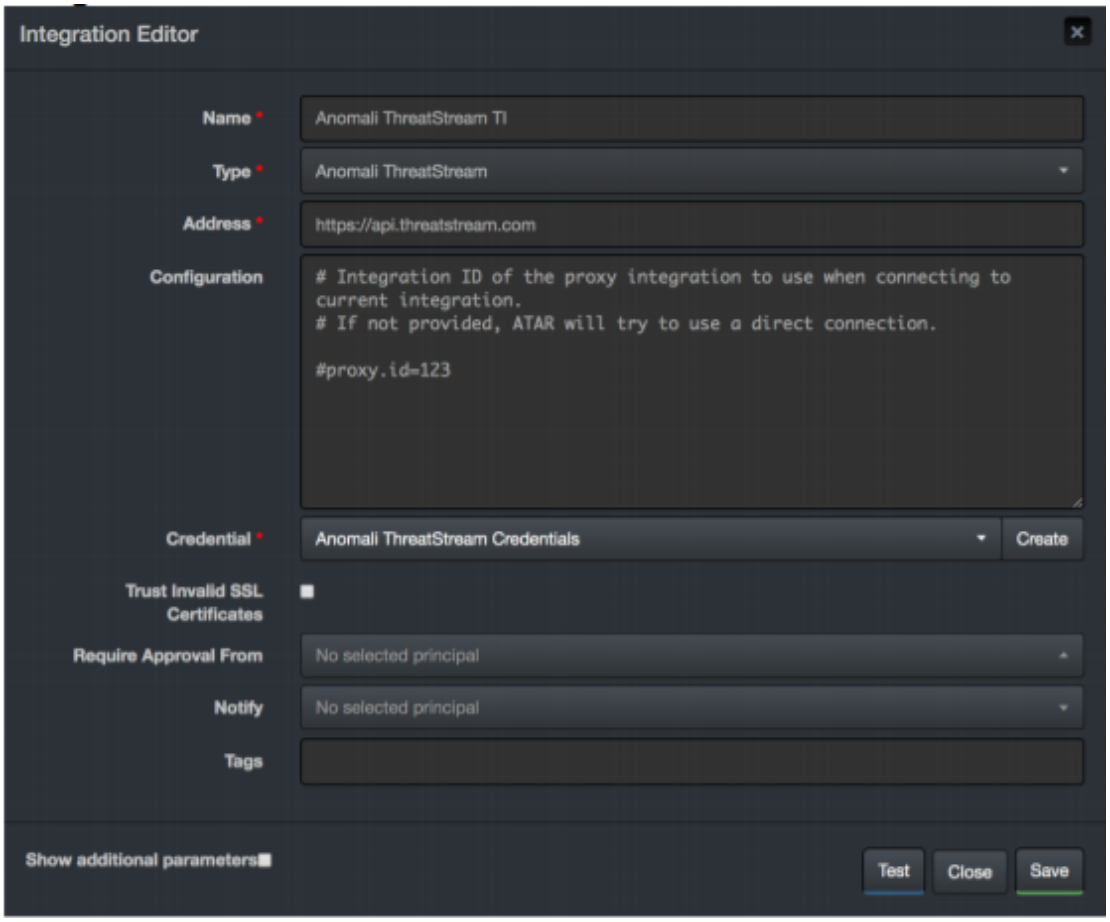
3. **Configuration** > **Integrations** > **Create Integration**.
4. Fill the configuration form with the following parameter values:

Parameter	Value
Name	Display name of Anomali ThreatStream integration on SOAR
Type	Anomali ThreatStream
Address	Address of the integration (https://api.threatstream.com).

Configuration	You need to specify the following configuration parameters:
	`# Integration ID of the proxy integration to use when connecting to`
	`# current integration.`
	`# If not provided, ATAR will try to use a direct connection.`
	`#proxy.id=123`

Credential	Name of the credential set you have just created on step 2. (For example, Anomali ThreatStream Credentials)
Trust Invalid SSL Certificates	No selection required
Require Approval From	Select user(s) from list to ask her/his approval before executing actions on this integration.
Notify	Select user(s) from the list to notify when SOAR performs an action on this integration.

.

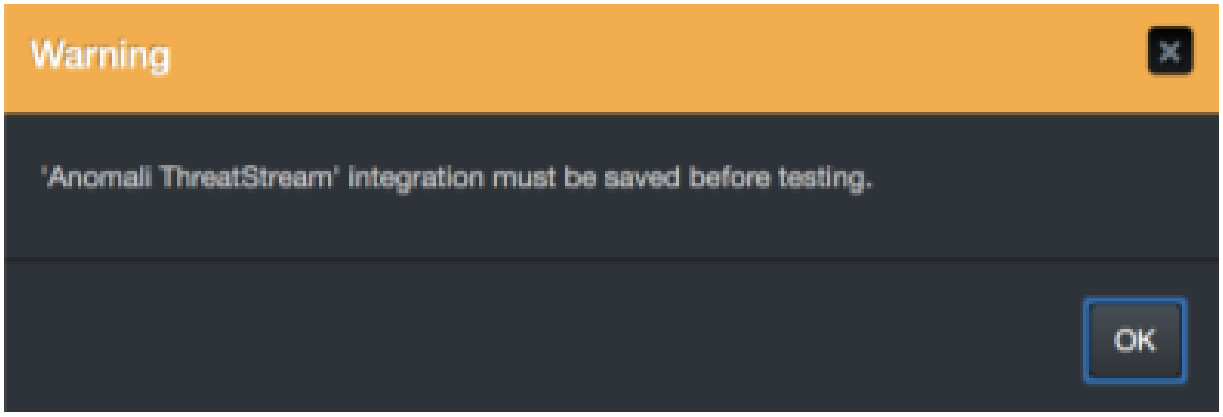


5. Click **Test**. The following pop up will be displayed if your credential and address are valid.

6. Click **Save** to complete integration.

Additoinal Notes

- Anomali ThreatStream integration on SOAR is an Advanced Script and content of the default script is accessible under **Configuration** > **Customization Library**.
- While defining the integration for the first time, you might encounter the following warning message, which is the expected behavior for this type of integration.



Integration Guide for Arbor Networks APS

Integration Overview

Arbor Networks APS is an in-line Distributed Denial of Service(DDoS) protection solution.

Integration Capabilities

ArcSight has the following integration capabilities with Arbor Networks APS:

- Block IP
- Block access to IP

Use Case: Blocking malicious IP on peripheral

ArcSight SOAR integrates with Arbor Networks APS to block malicious IP addresses detected while responding to an incident. SOAR can block both the incoming and outgoing traffic either automatically within a playbook or manually by an analyst.

Configuration

Prerequisites

- SOAR connects to Arbor Networks APS' API via HTTPS. By default, the API interface works on **443/tcp port**. So access permission to this port is required.
- An API access token needs to be created for SOAR to connect to Arbor Networks

APS.

Configuring Arbor Networks APS

1. Log in to Arbor Networks APS device.
2. Add a new API token.

```
admin@arbos: /# serv aaa local apitoken generate admin ATAR_INTEGRATION
```

```
Added token: jwP9JcmZYz4I9QH0LpkDA_n5nj_DNHifc6Iwsq0P
```

Note: SOAR uses the generated token as the credential password and user name as admin.

Configuring SOAR

1. Click **Configuration** > **Credentials** > **Create Credential**.
2. Fill the **Credential Editor form** with the following parameter values:

   a. **Internal Credential:**

Parameter	Value
Type	Internal credential
Name	Display name of the credential set (For example, Arbor APS Credential)
Username	admin
Password	API Token you have created for SOAR on Arbor Networks APS device
Private Key	Empty

   b. **Credential Store:**

Parameter	Value
Type	Extrenal credential
Name	Name of the credential with pull path of the safe on store

3. **Configuration** > **Integrations** > **Create Integration**.
4. Fill the configuration form with the following parameter values:

Parameter	Value
Name	Display name of Arbor Networks APS integration on SOAR
Type	Arbor Networks APS
Address	Address of the integration (the format should be http(s]://1.1.1.1:1234 or http[s]://abc.example.com:1234)
Password	API Token you have created for SOAR on Arbor Networks APS device

Credential	Name of the credential set you have just created on step 2. (For example, Arbor APS Credential)
Trust Invalid SSL Certificates	Select this if device's certificate is self-signed or not recognized by browsers
Require Approval From	Select user(s) from list to ask her/his approval before executing actions on this integration
Notify	Select user(s) from the list to notify when ATAR performs an action on this ntegration

5. Click **Test**.The following pop up will be displayed if your credential and address are valid.
6. Click **Save** to complete integration.

Integration Guide for Bind RPZ DNS

Integration Overview

ArcSight SOAR uses BIND DNS servers to block malicious domains using incident scope.

Integration Capabilities

Action

- Block

Configuration

Prerequisites

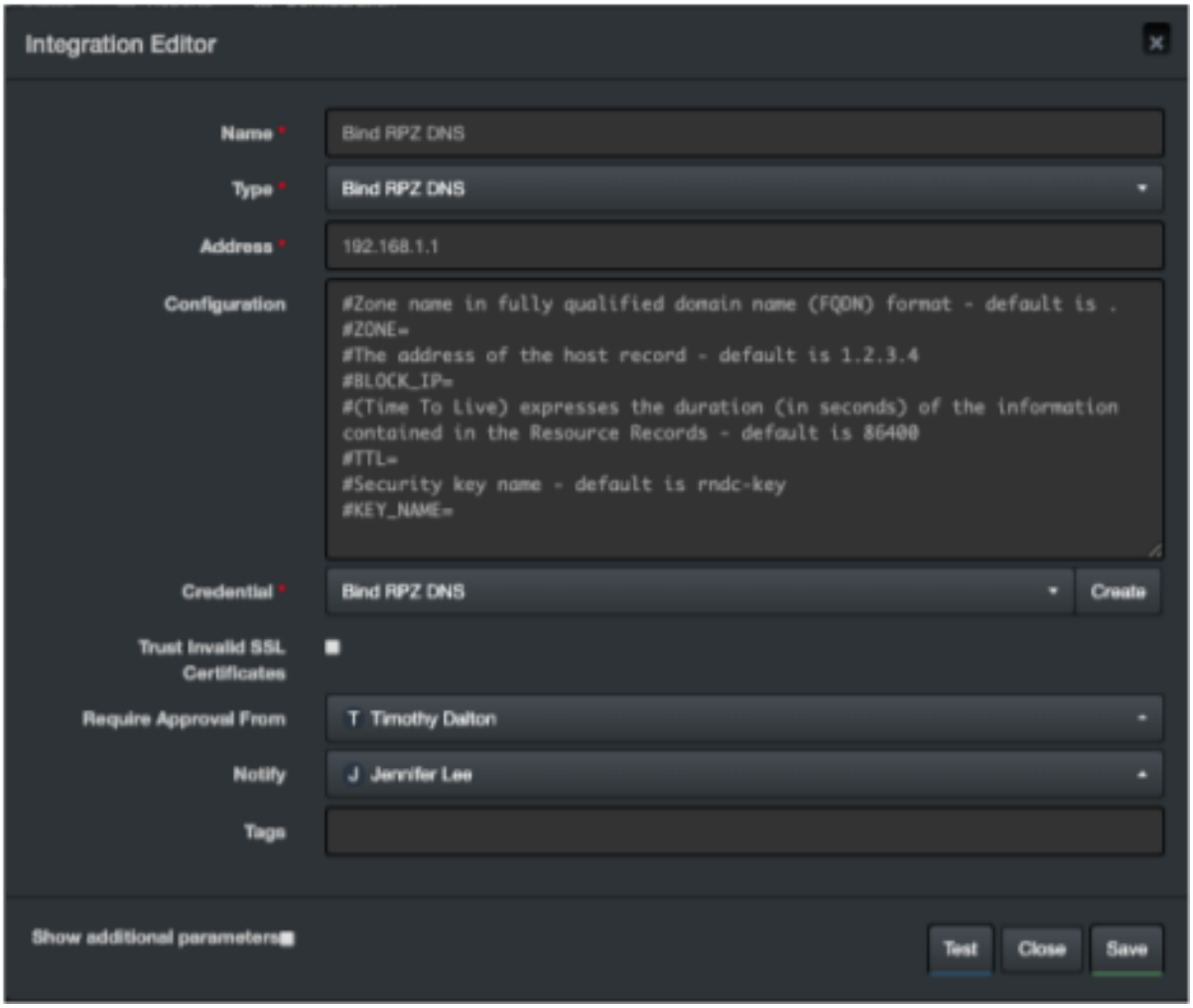
- You must enable the DNS Zone Transfer on the server as SOAR uses DNS Zone Transfer Protocol to connect to the BIND DNS server.
- Remote Name Daemon Control (RNDC)

Configuring SOAR

1. To create the integration, navigate to **Configuration** > **Integrations**.
2. Specify the following parameter values in the **Integration Editor window**:

Parameter	Value
Name	Display name of the integration
Type	BIND RPZ DNS
Address	Address of the integration (the format must be 1.1.1.1).

Configuration	You must specify the following configuration parameters: • **ZONE**: Name of the RPZ configured on the BIND server • **BLOCK_IP**: IP address to which malicious domains need to be redirected • **TTL**: Time-to-live for the DNS record • **KEY_NAME**: Name of the RNDC key
Credential	Specify the Credential that was defined for this integration under the **Credentials** menu
Trust Invalid SSL Certificates	Select this if Engine's certificate is self-signed or is not recognized by browsers.
Require Approval From	Select users from list who can provide approval before executing action on this integration
Notify	Select user(s) from the list to notify when SOAR performs an action on this integration



3. Click **Test**. The following pop up will be displayed if your credential and address are valid.
4. Click **Save** to complete integration.

Integration Guide for Carbon Black Response (EDR)

Integration Overview

Carbon Black Response (EDR)is a next-generation antivirus and end point detection response application. It's sophisticated detection combines custom and cloud-delivered threat intel, automated watchlists, and integrations with other platforms to efficiently scale hunt across the enterprise. It consolidates threat intelligence for your environment to automatically detect suspicious behavior.

Integration Capabilities

- Block Hash
- Unblock Hash
- Quarantine
- Unquarantine
- Computer Info
- Download Binary
- Get Binary Metadata
- List Process Connections
- Process Event Details
- Search Binaries
- Search Processes

Use Case: Investigating and Blocking Malware Spread

ArcSight SOAR integrates with Carbon Black Response (EDR), to help investigation and mitigation of malware attacks. When a suspicious file or malware is detected, SOAR lets you to search malware across endpoints, isolates PCs from network, and blocks relevant hashes. This investigation can either be performed automatically within a playbook or manually by an analyst.

Configuration

Prerequisites

- Access to port 443/tcp as SOAR connects to Carbon Black Response(EDR) API through HTTPS.
- An API key is required for SOAR to connect to Carbon Black Response(EDR).

Configuring Carbon Black Response(EDR)

1. Log in to Carbon Black Server.
2. Navigate to **User Profile** > **API Token** and make a note of the API key.

Configuring SOAR

1. Click **Configuration** > **Credentials** > **Create Credential**.
2. Specify the **Crenetial Editor** form with the following parameter values:
   a. **Internal credential:**

Parameter	Value
Type	Internal credential
Name	Display name of the credential set (For example, Carbon Black Credential)
Username	Empty
Password	Empty
Private Key	API Key obtained from Carbon Black Response (EDR).

   b. **Credential Store:**

Parameter	Value
Type	External credential
Name	Name of the credential with full path of the safe on store.

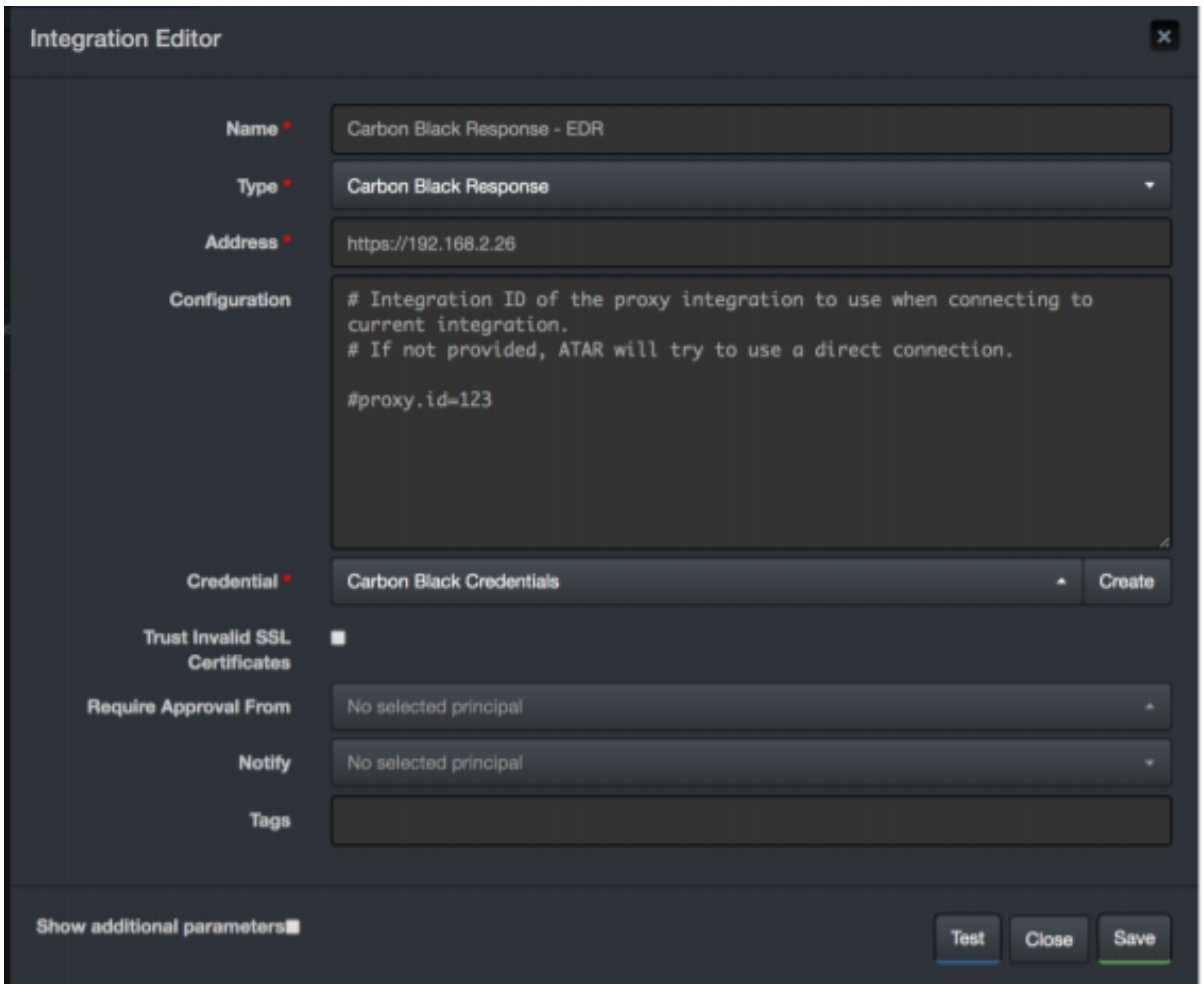
3. Click **Configuration** > **Integrations** > **Create Migration**.
4. Specify the **Configuration form** with the following parameter values:

Parameter	Value
Name	Display name of Carbon Black Response (EDR) integration on SOAR
Type	Carbon Black Response

Address	Address of the integration (in the format: https://192.168.2.26)

Configuration	Specify the following configuration parameters:
	`# Integration ID of the proxy integration to use when connecting to`
	`# current integration.`
	`# If not provided, SOAR will try to use a direct connection.`
	`#proxy.id=123`

Credential	Name of the credential set created on step 2. (For example, Carbon Black Credentials)
Trust Invalid SSL Cerificates	Not Applicable
Require Approval From	Select users from list who can provide approval before executing actions on this integration.
Notify	Select users from the list to notify when SOAR performs an action on this integration

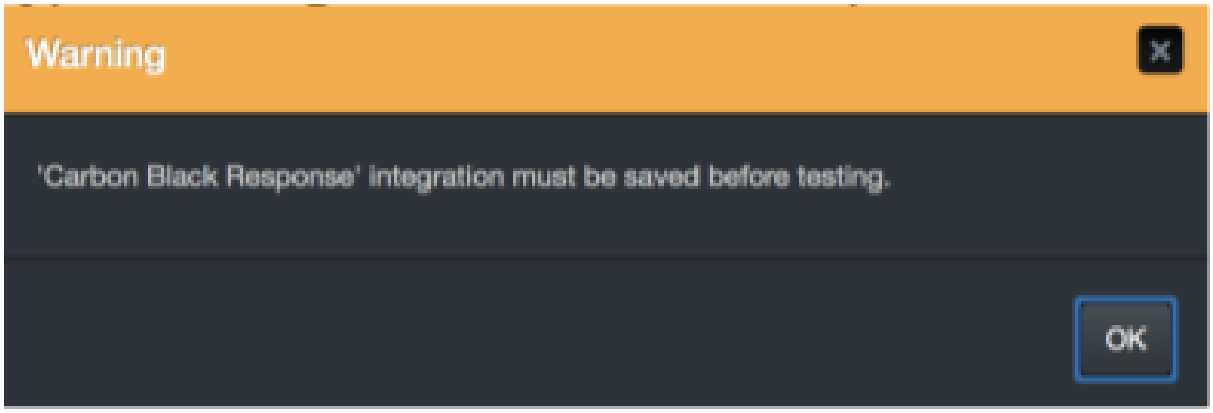


5. Click **Test**. The following pop up will be displayed if your credentials and address are valid.

6. Click **Save** to complete integration.

Additional Notes

- Carbon Black Response integration on SOAR is an Advanced Script, and the content of default script is accessible under **Configuration** > **Customization Library**.
- While defining the integration for the first time, you will encounter the following warning message, which is expected behavior for this type of integration.



Integration Guide for Check Point R80

Integration Overview

Check Point R80 Security Gateway is a firewall solution, which helps to maintain R80 integration over API. The API connection method is unavailable before R80 versions.

Integration Capabilities

Action

- Block Hash
- Block Host
- Block IP
- Block Email Sender
- Block URL

Configuration

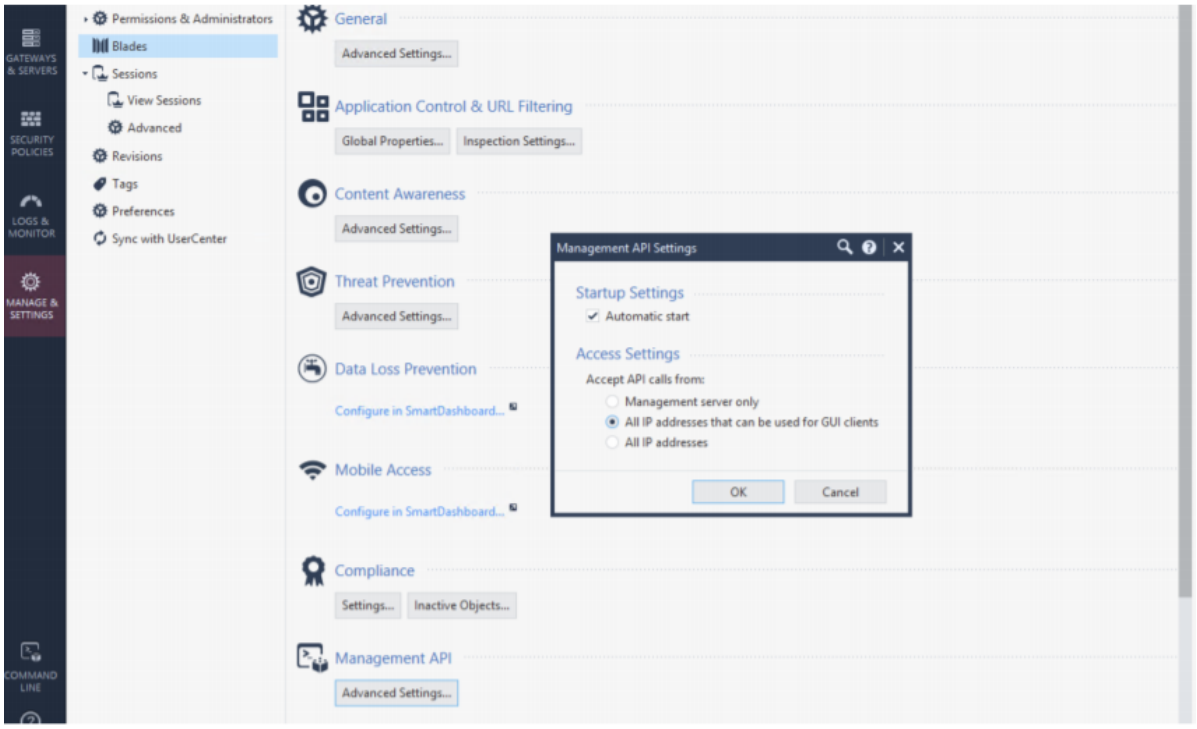
Configuring Check Point R80

1. To access the web API, change the default settings in the Management Console:

   Click **Manage&Settings** > **Blades** > **Management API Advanced Settings** > **Access Settings**

2. To send API requests from SOAR IP, select **All IP addresses that can be used for GUI clients** in the **Management API Settings** window.
3. To restart the API service, execute the following command in the command prompt:

   ```
   api restart
   ```



Configuring Access Rights

SOAR uses a standard read/write access for a user for the necessary policy and objects. Also, to install policy automatically, the users must have these rights in their permission profile. SOAR with full capabilities must have access to the following items:

Access Control

- Policy
- Data Loss Prevention
- Access Control Objects and Settings
- Install Policy

Threat Prevention

- Policy Layers
- Policy Exceptions
- Profiles
- Protections
- Install Policy

Others

Common Objects

Management

- Management API Login

Repeat the following steps to add each capability:

1. Click **Add Object** > **Publish** > **Install Policy**.

   Note: Install Policy step is optional, which can be selected in the integration configuration.

2. Add an **Object group** that will be used in SOAR configuration area.

Configuring SOAR

1. In **SOAR configuration**, provide the required Name, Address, and credential.

   Note: Configuration area is important in Check Point R80 configuration.

2. Specify the object group name created in Check Point Requirements step, such as:

   `group.name=SOAR`

3. Specify the **product types** AV (AntiVirus) for external threats and/or AB (AntiBot) for internal threats.

   `products=AV|AB`

   Note: For multiple products, use | as a separator. (AB product doesn't support Hash and E-Mail **actions**.

4. To install policy automatically, set **install.policy**variable from false to true.

   `install.policy=false`

5. Set the following parameters for the policy:

```
Provide policy package name used on Checkpoint.
policy.package=
Provide Gateways as targets for policy installation.
For multiple Gateways please use | as separator.
targets=
Select if you want to use access policy (true or false). Default false.
access=
Select if you want to use access policy (true or false). Default false.
threat.prevention=
```

6. Click **Configuration** > **Integrations**.
7. In the **Integrations Editor**, fill the following parameter values:

Parameter	Value
Name	Display name of the ntegration
Type	Check Point R80 Next Generation Firewall
Address	Address of the integration (the format should be 1.1.1.1 or http [s]://abc.example.com)

Configuration	Specify the following configuration parameters:
	`# Group name for adding object to block`
	`group.name=SOAR`
	`# Product types AV (AntiVirus for external threats) and/or AB`
	`# (AntiBot for internal threats)`
	`# Please put | separator for more than one product`
	`products=AV`
	`# If you would like to install policy automatically, please set`
	`# install.policy variable to true`
	`install.policy=false`
	`# Please set four variables below to install policy`
	`policy.package=`
	`# Please put | separator for more than one target`
	`targets=`
	`# true or false, false is default`
	`Soar v3.0 Page 4 of 4`
	`access=`
	`# true or false, false is default`
	`threat.prevention=`

Credential	Credential that has been defined for this integration under the Credentials menu
Trust Invalid SSL Certificates	Select this if Engine's certificate is self-signed or is not recognized by browsers.
Require Approval From	Select user(s) from list to ask her/his approval before executing actions on this integration
Notify	Select user(s) from the list to notify when SOAR performs an action on this integration

8. Click **Test**. The following pop up will be displayed if your credential and address are valid.
9. Click **Save** to complete the integration.

Integration Guide for Check Point SandBlast

Integration Overview

Check Point SandBlast provides advanced threat protection against known threats, zero-day malware, and sophisticated attacks.

Integration Capabilities

Threat Emulation capability prevents infections from undiscovered exploits, zero-day and targeted attacks by inspecting files, and running them in a virtual sandbox to discover malicious behavior.

ArcSight SOAR has the following integration capabilities with Check Point SandBlast:

- Threat Emulation & AV Scan

Use Case: Investigating suspicious file

With Check Point SandBlast integration, during the investigation of an incident, SOAR can send a suspicious file to Check Point SandBlast to emulate threats and run an anti virus scan for the file. This investigation can either be performed automatically within a playbook or manually by an analyst.

Configuration

Prerequisites

- Make sure you have access to 443/tcp port as SOAR connects to Check Point SandBlast's API through HTTPS. If cloud-based threat emulation service is used, the API interface works on https://te.checkpoint.com.
- If a local gateway is used, typically access permission to 18194/tcp port is required.
- An API key is required for SOAR to connect to Check Point SandBlast.

Configuring Check Point SandBlast

1. If you are using cloud-based threat emulation service, contact Check Point to get the API key.

2. If you are using local gateway, the following link provides you with the document for creating API key:

   http://supportcontent.checkpoint.com/solutions?id=sk113599

Configuring SOAR

1. **Configuration** > **Integrations** > **Create Integration**.
2. Fill the **Credential Editor** form with the following parameter values:
   a. **Internal Credential:**

Parameter	Value
Type	Internal Credential
Name	Display name of credential set (For example, Check Point SandBlast Credential)
Username	Empty
Password	Empty
Private Key	API key you have created for SOAR on local gateway or you have obtained from Check Point.

   b. **Credential Store:**

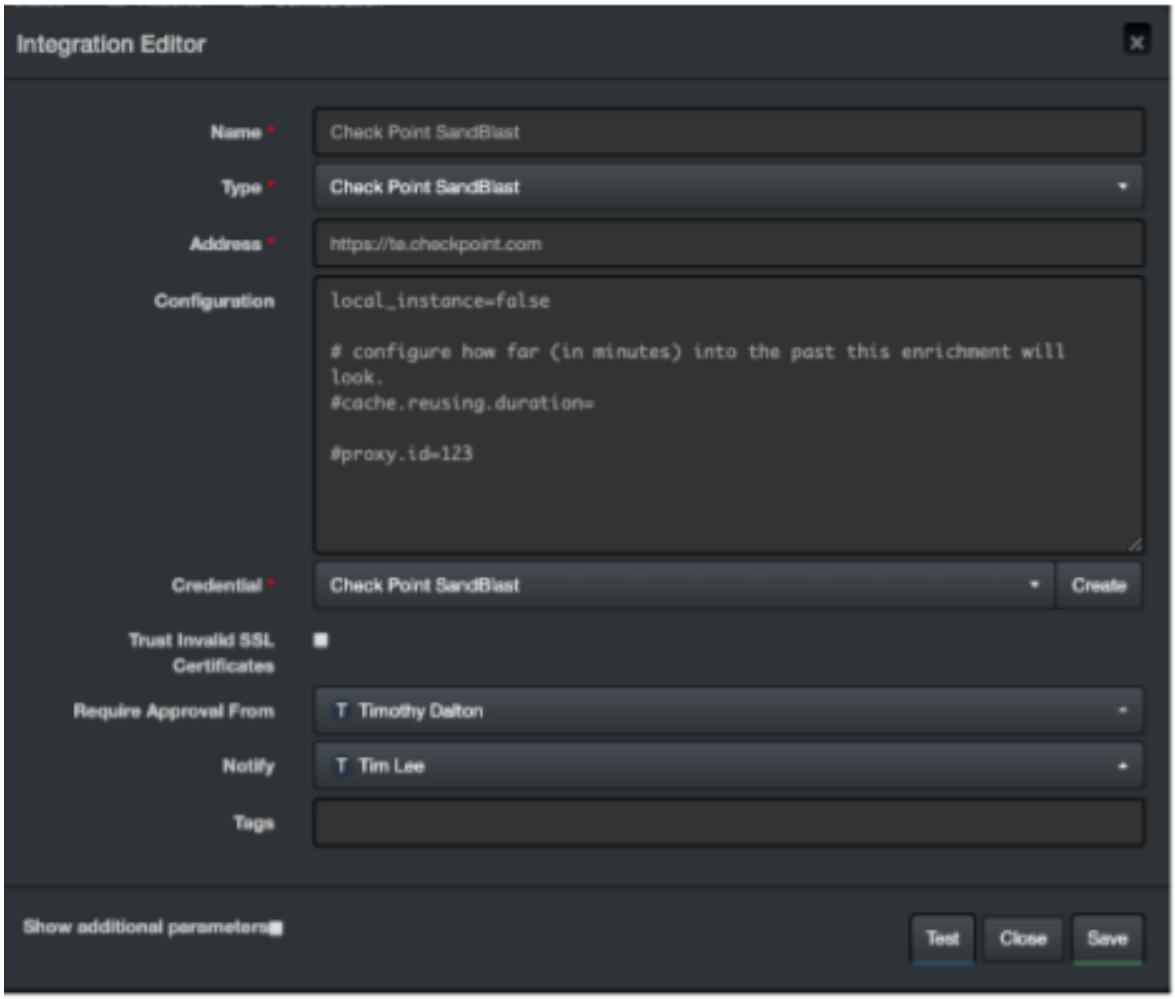
Parameter	Value
Type	External credential
Name	Name of the credential with full path of the safe on store

3. **Configuration** > **Integrations** > **Create Integration.**
4. Fill the configuration form with the following parameter values:

Parameter	Value
Name	Display name of Check Point SandBlast integration on SOAR
Address	Address of the integration (the format must be https://192.168.1.1:18194 or https://te.checkpoint.com)
Credential	Name of the credential set you have just created on step 2. (For example, Check Point SandBlast Credential).
Trust Invalid SSL Certificates	Select this if Engine's certificate is self-signed or is not recognized by browsers.

Configuration	Specify the following configuration parameters:
	`# Set local_instance true if you use local gateway.`
	`local_instance=false`
	`# configure how far (in minutes) into the past this enrichment will look.`
	`cache.reusing.duration=60`
	`# Set proxy id if necessary for SOAR to reach the SandBlast instance.`
	`proxy.id=123`

Require Approval Form	Select user(s) from list to ask her/his approval before executing actions on this s.
Notify	Select user(s) from the list to notify when SOAR performs an action on this integration.



5. Click **Test**. The following pop up will be displayed if your credential and address are valid.

6. Click **Save** to complete integration.

Integration Guide for Cisco ASA Firewall

Integration Overview

Cisco ASA is a security technology that combines firewall, antivirus, intrusion prevention, and virtual private network (VPN) capabilities. It provides proactive threat defense and stops attacks before they spread in the network.

Integration Capabilities

ArcSight SOAR has the following integration capabilities with Cisco ASA:

- Block IP
- Block FQDN

Use Case: Blocking access to malicious IP addresses and hosts

With this integration, SOAR can block malicious IP addresses and hosts on Cisco ASA firewall devices while responding to cyber-attacks. Blocking can be either performed automatically within a playbook or manually by an analyst.

Configuration

Prerequisites

- Make sure you have access to **443/tcp port** as SOAR connects to Cisco ASA Firewall's API through HTTPS. By default, Cisco ASA Firewall REST-API interface works on this port .
- Make sure SOAR has a user account to connect to Cisco ASA Firewall.

Configuring Cisco ASA Firewall

1. Log in to Cisco ASA Firewall device.
2. Create a user account with privilege level 15 as follows:

```
# configure terminal
```

```
# username soar password choose_a_complex_password privilege 15
```

3. To enable REST API, run the following commands:

```
# rest-api image
```

```
# rest-api agent
```

Configuring SOAR

1. Click **Configuration** > **Credentials** > **Create Credential**.
2. Fill the **Credential Editor** form with the following parameter values:
   a. **Internal Credential:**

Parameter	Value
Type	Internal Credential
Name	Display name of credential set (For example, Cisco ASA Credentials)
Username	User you have created for SOAR on Cisco ASA Firewall
Password	Password of the user you have created for SOAR on Cisco ASA Firewall.
Private Key	Empty

   b. **Credential Store:**

Parameter	Value
Type	External credential
Name	Name of the credential with full path of the safe on store

3. Click **Configuration** > **Integrations** > **Create Integration**.
4. Fill the configuration form with the following parameter values:

Parameter	Value
Name	Display name of Cisco ASA Firewall integration on SOAR
Type	Cisco ASA Firewall
Address	Address of the integration (the format must be https://192.168.1.1:18194).

Configuration	You must specify the following configuration parameters:
	`NETWORK_OBJECT_GROUP_NAME_FOR_IP=SOAR_BLOCK_IP`
	`NETWORK_OBJECT_GROUP_NAME_FOR_DOMAIN=SOAR_BLOCK_FQDN`
	`#proxy.id=2453`

Credential	Name of the credential set you have just created on step 2 (For example, Cisco ASA Credentials)
Trust Invalid SSL Certificates	Select this if Firewall's certificate is self-signed or is not recognized by browsers.
Require Approval Form	Select user(s) from list to ask her/his approval before executing actions on this integration.
Notify	Select user(s) from the list to notify when SOAR performs an action on this integration.

5. Click **Test**. The following pop up will be displayed if your credential and address are valid.
6. Click **Save** to complete integration.

Additional Notes

- Cisco ASA Firewall integration on SOAR is an Advanced Action Script, and you can access the content of the default script under **Configuration** > **Customization Library**.
- While defining integration for the first time, you might encounter the following warning message, which is the expected behavior for this type of integration.
- If the Object Group Names you have created do not exist in **Configuration**, SOAR automatically creates them.
- For physical ASAs, the REST API Agent is published separately from other ASA images and does not include the REST API plug-in package. You must download the REST API package on the device's flash and install it using the **rest-api image** command.
- The deployment package with a virtual ASA (ASAv) includes the REST API image. However, its default configuration does not allow you to use the REST API.

  For information, see Cisco ASA REST API Quick Start Guide:
  https://www.cisco.com/c/en/us/td/docs/security/asa/api/qsg-asa-api.html

Integration Guide for Cisco Firepower Management Center

Integration Overview

Cisco Firepower Management Center (formerly Sourcefire Firepower Management Center) is an administrative center node of the Firepower Threat Defense systems and manages critical Cisco network security solutions. It provides complete and unified management over firewalls, application control, intrusion prevention, URL filtering, and advanced malware protection.

Integration Capabilities

ArcSight SOAR has the following integration capabilities with Cisco Firepower Management Center:

- Block IP Address
- Block URL

Use Case: Blocking access to malicious IPs & URLs

With this integration, SOAR can block malicious IP addresses and URL addresses on multiple firewall devices simultaneously while responding to cyber-attacks. This blocking can be either performed automatically within a playbook or manually by an analyst.

Configuration

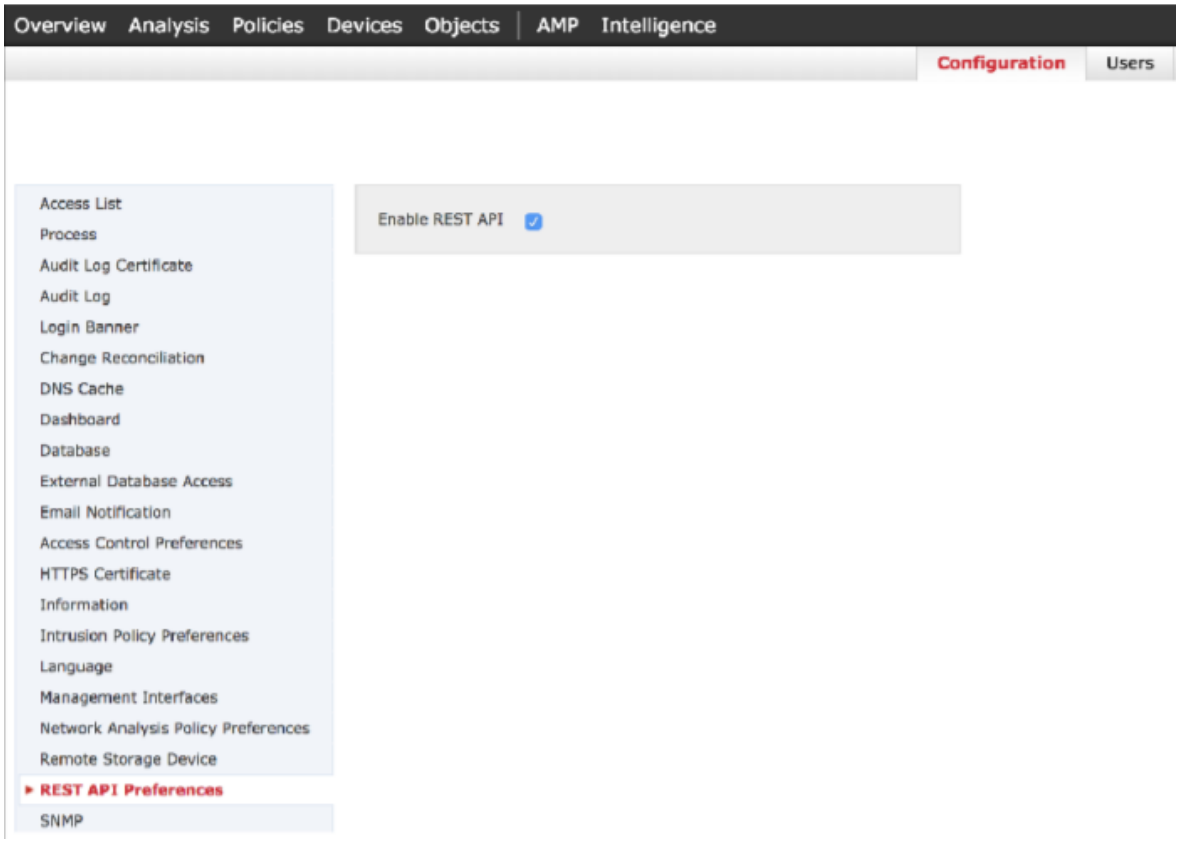
Prerequisites

Make sure you have the following prerequisites:

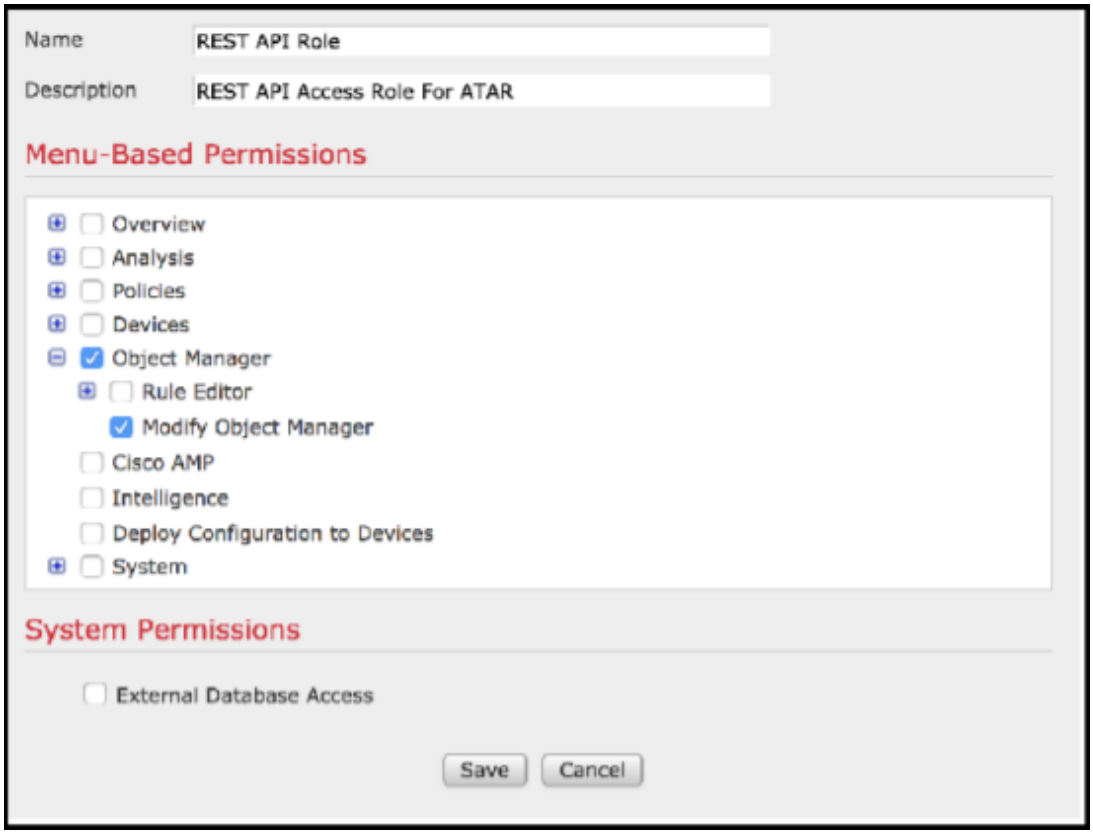
- Access to **443/tcp port**as SOAR connects to Cisco Firepower Management Center API through HTTPS.
- The current version of Cisco Firepower Management Center 6.2.3 (build 83), as SOAR supports it.
- User account for SOAR to connect to Cisco Firepower Management Center.

Configuring Cisco Firepower Management Center

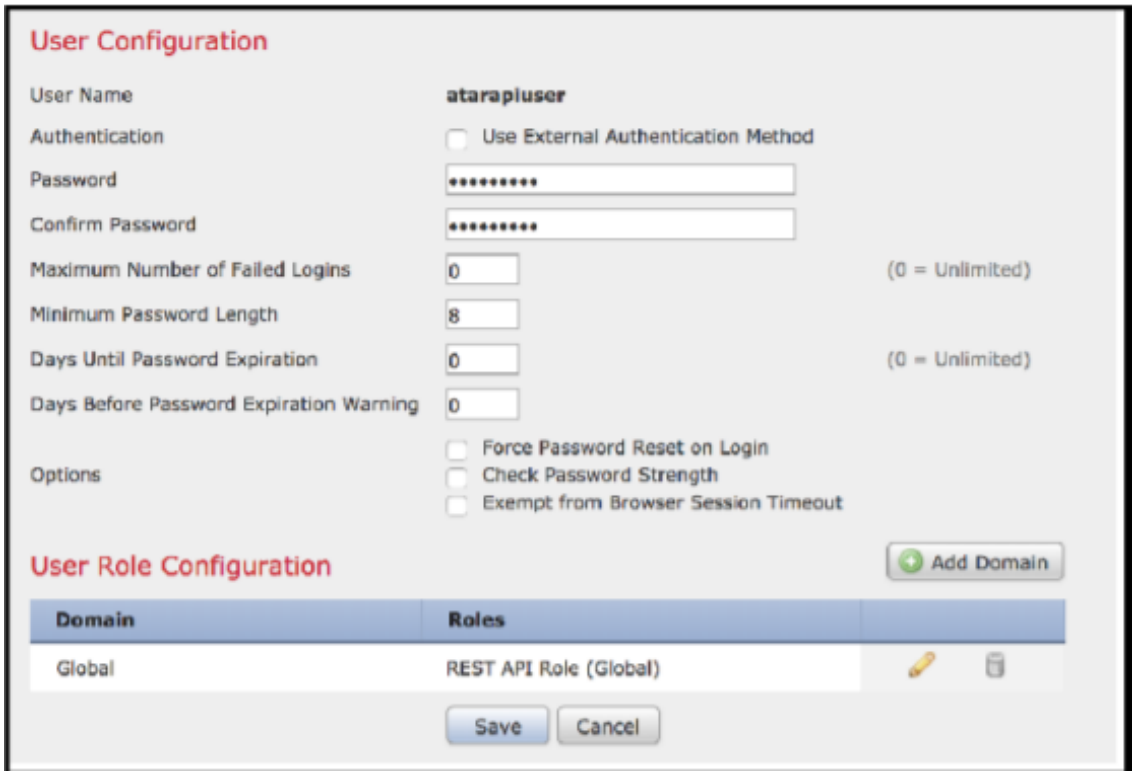
1. Navigate to **System** > **Configuration** > **REST API Preferences**.
2. Select **Enable REST API**, if it is not enabled.



3. To create a role for SOAR, navigate to **System** > **Users** > **User Roles**.
   a. Select **Modify Object Manager** for permission access.
   b. (Optional) If automatic deployment of the configuration changes are required, then you must select **Deploy Configuration to Devices** permission.

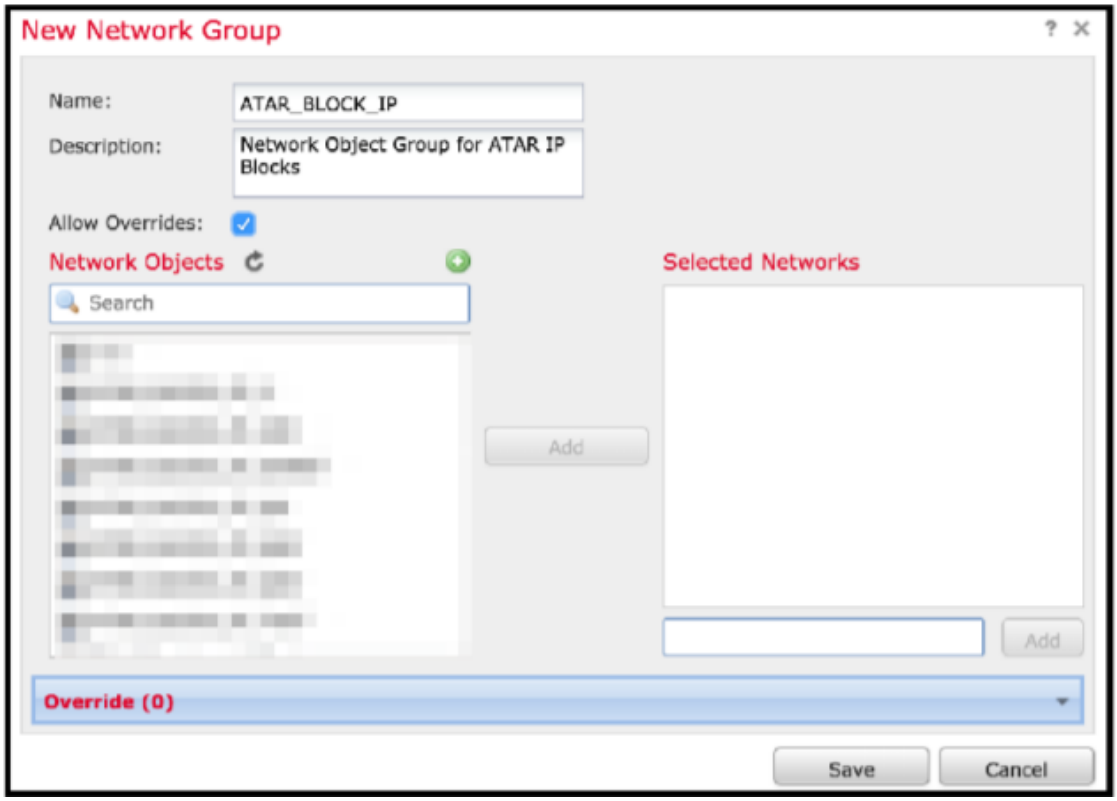


4. Navigate to **System** > **Users** > **Users**.
5. Create a user with user role that you have created in the previous step.



6. To add two groups: one for IP addresses and another group for URLs for SOAR to manage, navigate to **Objects** > **Object Management**.

Note: You can use these object groups in required rules.



Configuring SOAR

1. Click **Configuration** > **Credentials** > **Create Credential**.
2. Fill the Credential Editor form with the following parameter values:

a. **Internal Credential:**

Parameter	Value
Type	Internal Credential
Name	Display name of credential set (For example, Cisco Firepower Management Credentials)
Username	User you have created for SOAR on Cisco Firepower Management Center
Password	Password of the user that you have created for SOAR on Cisco Firepower Management Center.
Private Key	Empty

b. **Credential Store:**

Parameter	Value
Type	External Credential
Name	Name of the credential with pull path of the safe on store.

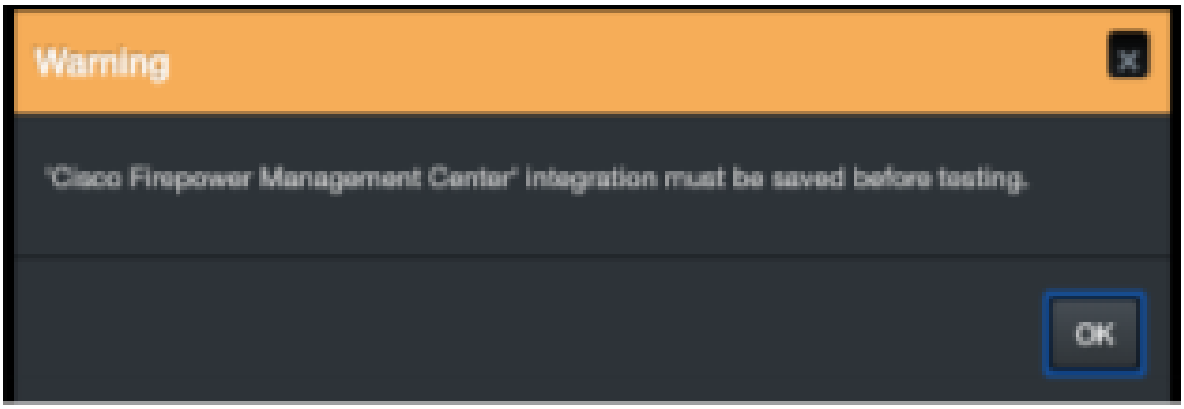
3. Click **Configuration** > **Integrations** > **Create Integration**.
4. Fill the configuration form with the following parameter values:

Parameter	Value
Name	Display name of Cisco Firepower Management Center integration on SOAR
Type	Cisco Firepower Management Center
Address	Address of the integration (the format must be https://192.168.2.3)
Credential	Name of the credential set you have just created on step 2 (For example, Cisco Firepower Management Credentials)
Trust Invalid SSL Certificates	Select this if Firewall's certificate is self-signed or not recognized by browsers
Configuration	You must specify the following configuration parameters. `NETWORK_OBJECT_GROUP_NAME = SOAR_BLOCK_IP URL_OBJECT_GROUP_NAME = SOAR_BLOCK_URL`
Require Approval From	Select user(s) from list to ask her/his approval before executing actions on this integration
Notify	Select user(s) from the list to notify when SOAR performs an action on this integration.

5. Click **Test**. The following pop up will be displayed if your credential and address are valid.
6. Click **Save** to complete integration.

Additional Notes

- Cisco Firepower Management Center integration on SOAR is an Advanced Action Script, and you can access the content of the default script under **Configuration** > **Customization Library**.
- While defining integration for the first time, you might encounter the following warning message, which is the expected behavior for this type of integration.



Integration Guide for Cisco Identity Service Engine

Integration Overview

The Cisco Identity Services Engine (ISE) offers a network-based approach for adaptable, trusted access everywhere, based on the context. It provides intelligent, integrated protection through intent-based policy and compliance solutions.

Integration Capabilities

ArcSight SOAR has the following integration capability with Cisco Identity Services Engine:

Action:

- Block MAC Address

Configuration

Prerequisites

Make sure to check the following prerequisites:

- Current version of Cisco Identity Services Engine 2.3.0.238 as SOAR supports it.
- Access to 443/tcpport as SOAR connects to Identity Services Engine API through HTTPS.
- An user account for SOAR to connect to Identity Services Engine

Configuring Cisco Identity Services Engine

1. Create a user account and the user must be a member of MnT Admin.

Configuring SOAR

1. Click **Configuration** > **Credentials** > **Create Credential**
2. Fill the **Credential Editor** form with following parameter values:

a. **Internal Credential:**

Parameter	Value
Type	Internal Credential
Name	Display name of credential set (For example, Cisco ISE credentials)
Username	User you have created for SOAR on Cisco Identity Services Engine
Password	Password of the user that you have created for SOAR on Cisco Identity Services Engine.
Private Key	Empty

b. **Credential Store:**

Parameter	Value
Type	External Credential
Name	Name of the credential with pull path of the safe on store.

3. Click **Configuration** > **Integrations** > **Create Integration**.
4. Fill the configuration form with the following parameter values:

Parameter	Value
Name	Display name of Cisco Identity Services Engine integration on SOAR
Type	Cisco Identity Services Engine
Address	Address of the integration (the format must be https://192.168.2.3)
Credential	Name of the credential set you have just created on step 2 (For example, Cisco ISE Credentials)
Trust Invalid SSL Certificates	Select this if Firewall's certificate is self-signed or is not recognized by browsers
Configuration	You must specify the following configuration parameters. `serverHost =`
Require Approval From	Select user(s) from list to ask her/his approval before executing actions on this integration
Notify	Select user(s) from the list to notify when SOAR performs an action on this integration.

5. Click **Test**. The following pop up will be displayed if your credential and address are valid.
6. Click **Save** to complete integration.

Integration Guide for Cisco Ironport Email Security

Integration Overview

Cisco Ironport Email Security is one of Cisco Ironport products to prevent phishing, business e-mail compromise, ransomeware and spam.

Integration Capabilities

ArcSight SOAR has the following integration capabilities with Cisco Ironport Email Security:

- Block sender IP/Host
- Block email that includes a keyword
- Block sender email

Use Case: Stopping phishing campaigns

With this integration, SOAR can block emails based on sender, IP address or a keyword while responding to cyber-attacks. Blocking can be either performed automatically within a playbook or manually by an analyst.

Configuration

Prerequisites

Make sure to check the following prerequisites:

- Current version of Cisco Ironport Email Security 11.0.0-264 as SOAR supports it.
- Access to 22/tcp port as SOAR connects to Cisco Ironport Email Security via SSH.
- A user account for SOAR to connect to Cisco Ironport Email Security.

Configuring Cisco Ironport Email Security

1. To access the **Cisco Inroport Email Security resources**, create a user account with minimum **operator** role.

Configuring SOAR

1. Click **Configuration** > **Credentials** > **Create Credential**.
2. Fill the Credential Editor form with the following parameter values:
   a. **Internal Credential:**

Parameter	Value
Type	Internal Credential
Name	Display name of credential set (For example, Cisco Firepower Management Credentials)
Username	User you have created for SOAR on on Cisco Firepower Management Center
Password	Password of the user that you have created for SOAR on Cisco Firepower Management Center.
Private Key	Empty

   b. **Credential Store:**

Parameter	Value
Type	External Credential
Name	Name of the credential with pull path of the safe on store.

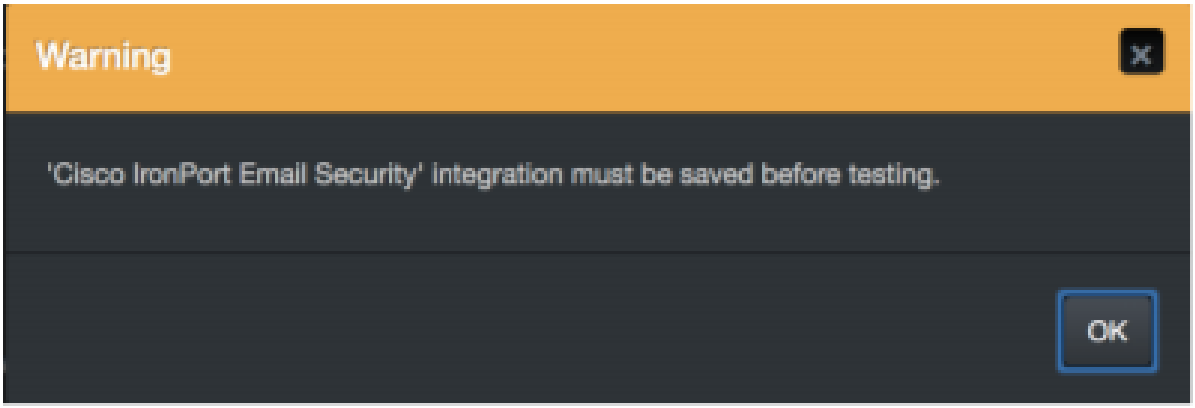
3. Click **Configuration** > **Integrations** > **Create Integration**.
4. Fill the configuration form with the following parameter values:

Parameter	Value
Name	Display name of Cisco Ironport Email Security integration on SOAR
Type	Cisco Ironport Email Security
Address	Address of the integration (the format must be 192.168.200.43)
Credential	Name of the credential set you have just created on step 2 (For example, Cisco Ironport Credentials)
Require Approval From	Select user(s) from list to ask her/his approval before executing actions on this integration
Notify	Select user(s) from the list to notify when SOAR performs an action on this integration.

5. Click **Test**. The following pop up will be displayed if your credential and address are valid.
6. Click **Save** to complete integration.

Additional Notes

- Cisco Ironport Email Security integration on SOAR is an Advanced Action Script, and you can access the content of the default script under **Configuration** > **Customization Library**.
- While defining integration for the first time, you might encounter the following warning message, which is the expected behavior for this type of integration.



Integration Guide for Cyberark Central Credential Provider

Integration Overview

CyberArk Application Identity Manager is a central credential provider that stores passwords and other credentials used by systems, applications, and scripts by eliminating embedded credentials. SOAR might use encrypted credentials stored on its database and CyberArk AIM vault to connect to other systems and applications while investigating and responding to an incident.

Configuration

Prerequisites

- Make sure to check the access to CyberArk Application Identity Manager API as SOAR connects to it through HTTPS.
- Define a **new application for SOAR** on CyberArk's PVWA (Password Vault Web Access) Interface.

Configuring CyberArk Application Identity Manager

1. Log in to **Password Vault Web Access** interface as a user with **Manage Users** authorization permission.
2. Navigate to **Applications**and click **Add Application**.
3. Fill the Add Application form with the following parameter values:

Parameter	Value
Name	Specify SOAR as the unique name (ID) of the application.

Description	Specify a short description of the application (For example, Application for Automated Threat Analysis&Response)
Business Owner	Specify contact information about the application's Business owner
Location	Specify the location of the application in the Vault hierarchy. **Note**: If the location is not selected, the application gets added to the user location who creates it.

4. To specify unlimited number of machines and Windows OS users for a single application, select **Allow extended authentication restrictions.**
5. Navigate to **Allowed Machine**and specify the application's Allowed Machines.

Note: information enables the Credential Provider to check only applications that run from specified machines can access their passwords.

Configuring SOAR

1. Click **Configuration** > **Credentials** > **Create Credential**.
2. Fill the **Credential Editor** form with the following parameter values:
   a. **Internal Credential**:

Parameter	Value
Type	Internal Credential
Name	Display name of credential set (For example, CyberArk AIM Credential)
Username	Application Name you have created on CyberArk Password Vault Web Access
Password	Empty
Private Key	Empty

3. Click **Configurations** > **Integrations** > **Create Integration**.
4. Fill the **Configuration** form with the following details:

Parameter	Value
Name	Display name of CyberArk AIM integration on SOAR
Type	CyberArk Central Credential Provider

Address	Address of the integration (the format must be https://192.168.1.1:1234 or https://abc.example.com:1234)
Credential	Name of the credential set you have just created on step 2 (For example, CyberArk AIM Credential).
Trust Invalid SSL Certificates	Select this if device's certificate is self-signed or is not recognized by browsers
Require Approval From	Select user(s) from list to ask her/his approval before executing actions on this integration
Notify	Select user(s) from the list to notify when SOAR performs an action on this integration

5. Click **Test**. The following pop up will be displayed if your credential and address are valid.
6. Click **Save** to complete integration.

Additional Notes

Following are the steps to use CyberArk AIM as central credential store:

1. Navigate to **Configuraiton** > **Parameters**.
2. Modify the **ExternalCredentialStoreIntegrationID** parameter value to ID of the CyberArk AIM integration that you have defined in the above procedure.
3. To define the new name for a credential:
   a. Navigate to **Configuration** > **Credentials**.
   b. Select External Credential from the drop down and it automatically uses CyberArk AIM integration.

Note: The name of the credential must be the same as the account name defined in CyberArk. Make sure to follow the naming convention of SOAR as Safe and Folder separated by **|** character. Else, SOAR automatically searches all Safes for the given credential name.

Integration Guide for CYMRU Malware Hash Registry Query

Integration Overview

CYMRU is a look-up service that checks if the hash code is malware. If the hashcode belongs to malware, then the latest timestamp of the malware and the rough antivirus package detection rate is returned. ArcSight SOAR uses CYMRU Malware Hash Registry Query to query computed MD5 or SHA-1 hash of a file to check for malware.

Integration Capabilities

Action

- Hash registry query

Configuration

Configuring CYMRU Malware Hash Registry Query

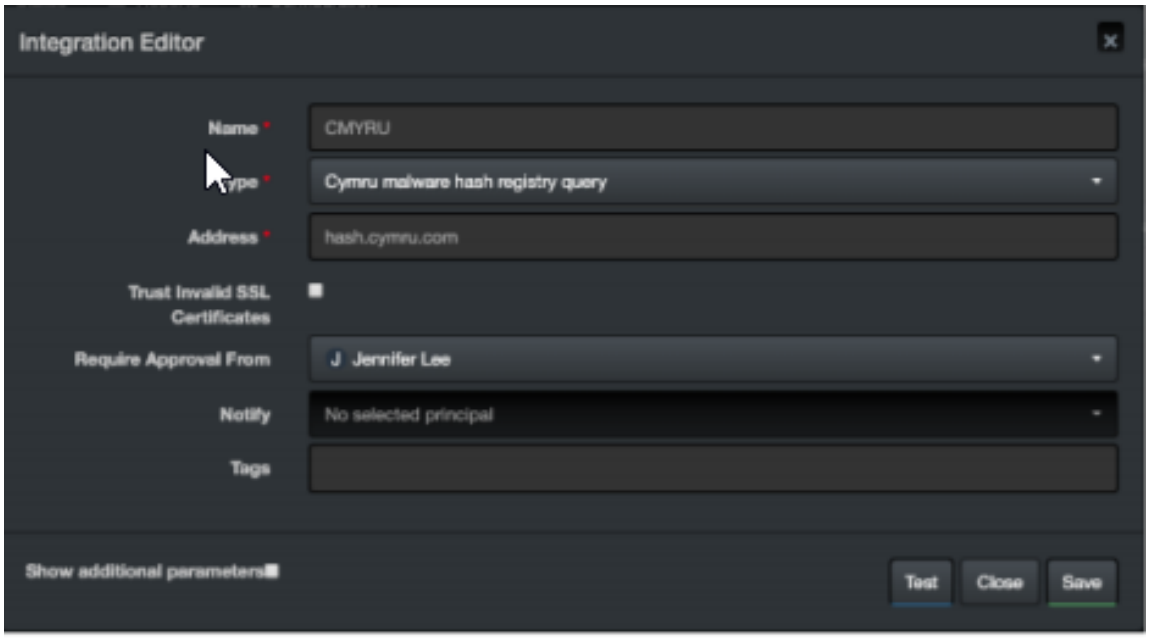
1. Make sure SOAR has access to CYMRU Malware Hash Registry Query integration's API as it connects to it through HTTPS.

Configuring SOAR

1. To create the integration, navigate to **Configuration** > **Integrations**.
2. Specify the following parameter values in the **Integrations Editor**:

Parameter	Value
Name	Display name of the integration
Type	CYMRU malware hash registry query
Address	Address of the integration (in the following format http[s]://malware.cymru.hash.com)

Trust Invalid SSL Certificates	Select this if Engine's certificate is self-signed or is not recognized by browsers.
Require Approval From	Select users from the list who can provide approval before executing actions on this integration
Notify	Select users from the list to notify when SOAR performs an action on this integration.



3. Click **Test** to test the integration.
4. Click **Save** to complete the integration.

Integration Guide for CyThreat Threat Intelligence

Integration Overview

CyThreat Threat Intelligence provides feeds with detailed analysis in the following categories:

- IP Risk List
- Hash Risk List
- Domain Risk List
- USOM Black

Integration Capabilities

ArcSight SOAR has the following integration capabilities with CyThreat Threat Intelligence Ingest Feed as Alert.

Use Case: Blocking malicious URLs and IPs before they harm

SOAR integrates with CyThreat intelligence feed and helps to block malicious entities on your perimeter protection before they harm.

Configuration

Prerequisites

1. Make sure SOAR has access to https://cti.stm.com.tr/api/ as SOAR connects to CyThreat intelligence feed through HTTPS.
2. API token and password to connect to CyThreat Threat intelligence API

Configuring CyThreat Threat Intelligence

No specific configurations are required on CyThreat Threat Intelligence.

Configuring SOAR

Configuring Credentials

1. Click **Configuration** > **Credentials** > **Create Credential**.
2. Specify the **Credential Editor** with the following parameter values:
   a. **Internal Credential**

Parameter	Value
Type	Internal Credential
Name	Display name of credential set (For example, CyThreat API Credentials)
Username	Empty
Password	API Password obtained from CyThreat
Private Key	API Token obtained from CyThreat

   b. **Credential Store:**

Parameter	Value
Type	External Credential
Name	Name of the credential with pull path of the safe on store.

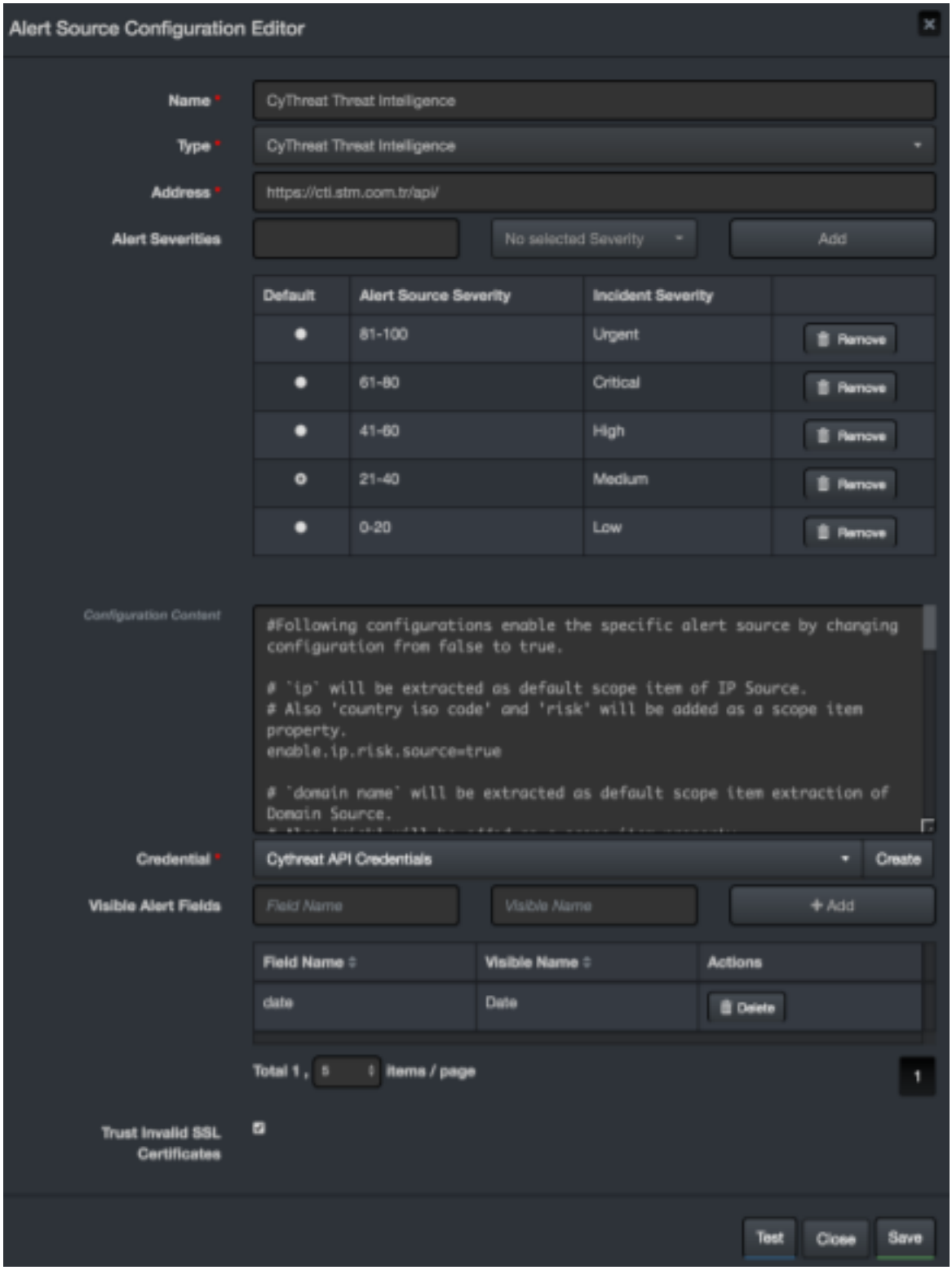
Configuring CyThreat Threat Intelligence Feed as Alert Source

1. Click **Configuration** > **Alert Source** > **Create Alert Source Configuraiton**.
2. Specify the following parameter values in the **Configuration** form:

Parameter	Value
Name	Display name of CyThreat Threat Intelligence Feed Alert Source on SOAR
Type	CyThreat Threat Intelligence
Address	Address of CyThreat Threat Intelligence (in the format https://cti.stm.com.tr/api/).
Alert Severities	Mapping of alert severity values to SOAR incident severities

Configuration	Specify the following configuration parameter values:
	`#Following configurations enable the specific alert source by changing configurat`
	`# `ip` will be extracted as default scope item of IP Source.`
	`# Also 'country iso code' and 'risk' will be added as a scope item property.`
	`enable.ip.risk.source=true`
	`# `domain name` will be extracted as default scope item extraction of Domain Source.`
	`# Also 'risk' will be added as a scope item property.`
	`enable.domain.risk.source=true`
	`# `hash` will be extracted as default scope item of Hash Source.`
	`# Also 'algorithm' and 'risk' will be added as a scope item property.`
	`enable.hash.risk.source=true`
	`# `url` will be extracted as default scope item of USOM source.`
	`# Note that severity mapping doesn't work for USOM source. Because there is no risk value of USOM records.`
	`# USOM severity will be the default value of alert source severity mapping.`
	`enable.usom.blacklist.source=true`
	`# Minimum risk threshold to get the alarm in. If not set, SOAR will process every alarm.`
	`# Value must be between 0-100 (lowest to highest).`
	`ip.min.risk=60`
	`domain.min.risk=60`
	`hash.min.risk=60`
	`# Integration ID of the proxy integration to use when connecting to current integration.`
	`# If not provided, SOAR will try to use a direct connection.`
	`#proxy.id=5422`
	`# How far (in days) into the past SOAR will look for remote incidents at the init`
	`# If not provided, SOAR will use full set of the source.`
	`# Note that STM CyThreat date filter accepts only 14 days.`

	days.to.look.back.at.initial.sync=14 # Note: Field names must start with / character # # Example: correlated.scope=/ip:NETWORK_ADDRESS:OFFENDER, /url:URL:OFFENDER # correlated.scope=
Credential	Name of the credential set created (For example, CyThreat API Credentials).
Trust Invalid SSL Certificates	Unselect
Visible Alert Fields	Define which alarm fields to be displayed on Incident Management Service Desk



3. Click **Test**. The following pop up will be displayed if your credential and address are valid.
4. Click **Save** to complete integration.

Additional Notes

- SOAR VM contains JDBC drivers for PostgreSQL on its filesystem. For Oracle, MySQL and Microsoft SQL Server, you must download respective JDBC drivers and copy them to SOAR VM.
- SQL queries which SOAR uses are defined under **Configuration** > **Customization** Library. Here, you must create new customizations with SQL Query type.

Integration Guide for DNS Service

Integration Overview

DNS Server is used to resolve and translate the IP addresses, host names and queries to various DNS records.

Integration Capabilities

SOAR has the following integration capabilities with DNS Server.

- DNS Lookup

Configuration

Prerequisites

- Make sure SOAR has access to DNS Server through 53/udp port

Configuring DNS Service

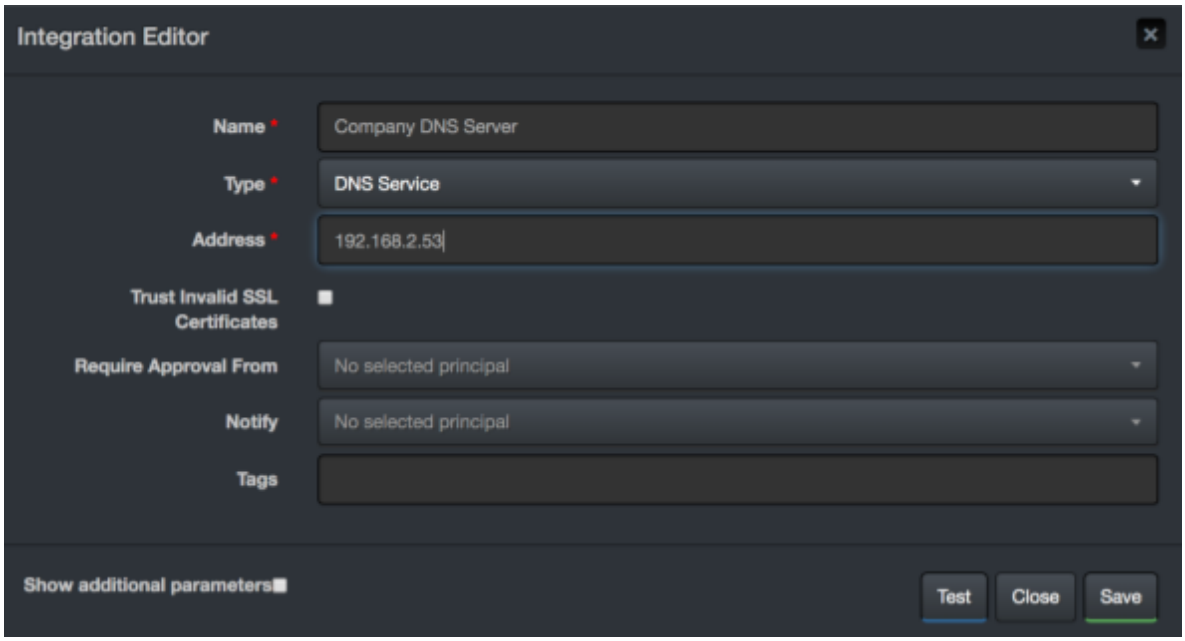
- No specific configuration is needed on DNS Server.

Configuring SOAR

1. Click **Configuration** > **Integrations** > **Create Integrations**.
2. Specify the following parameter values in the **Configuration** form:

Parameter	Value
Name	Display name of DNS Server integration on SOAR.
Type	DNS Service
Address	Address of the integration (in the format: 192.168.2.53)

Trust Invalid SSL Certificates	Not applicable
Require Approval From	Select users from the list who can provide approval before executing actions on this integration. As SOAR only executes enrichment on DNS Server, leave it empty
Notify	Select users from the list to notify when SOAR performs an action on this integration. As SOAR only executes enrichment on DNS Server, leave it empty

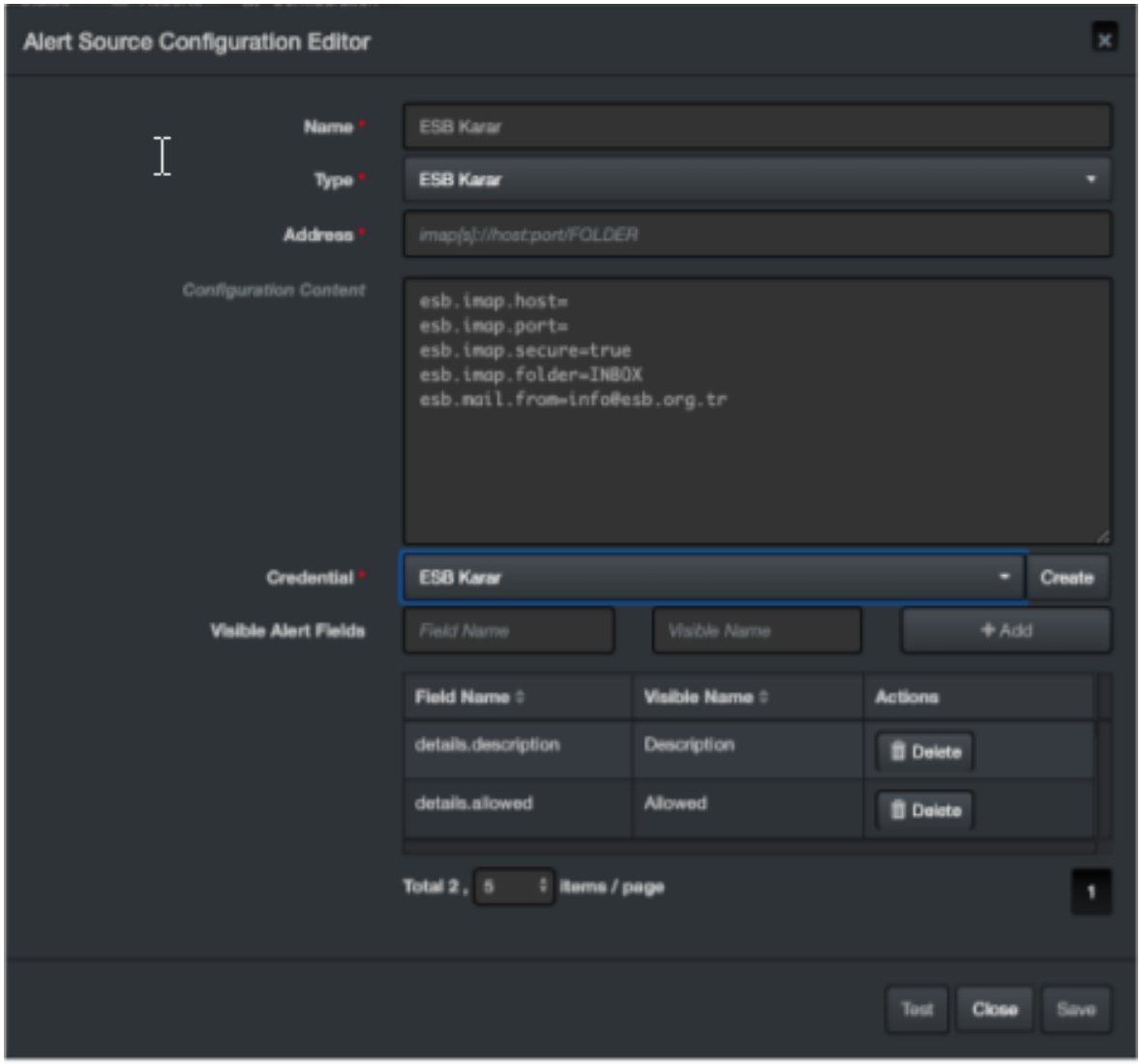


3. Click **Test**. The following pop up will be displayed if your credential and address are valid.
4. Click **Save** to complete integration.

Integration Guide for ESB Karar

1. To create the alert source, click **Configuration** > **Alert Source**.
2. Specify the following parameter values in the **Configuration Editor**:

Parameter	Value
Name	Display name of the alert source
Type	ESB Karar
Address	Address of the alert source. (in the format imap[s]://host:port/FOLDER).
Configuration Content	esb.imap.host= esb.imap.port= esb.imap.secure=true esb.imap.folder=INBOX esb.mail.from=info@esb.org.tr
Credential	Credential defined for this alert source under the Credentials menu
Visible Alert Field	- details.description - details.allowed



3. Click **Test**. The following pop up will be displayed if your credential and address are valid.
4. Click **Save** to complete integration.

Integration Guide for F5 Big-IP Advanced Firewall Manager

Integration Overview

Big IP AFM protects the network against incoming threats, even the most massive and complex DDoS attacks.

Big IP AFM keeps bad traffic away from some specific network addresses and protects the data center against DDoS attacks, and other network or application attacks. It also brings visibility and control to SSH, and SSL connections, providing against back door threats that use the SSH channel for data breaches and app attacks.

Integration Capabilities

Action

- Add address to specific address list

Configuration

Configuring F5 Big-IP Advanced Firewall Manager

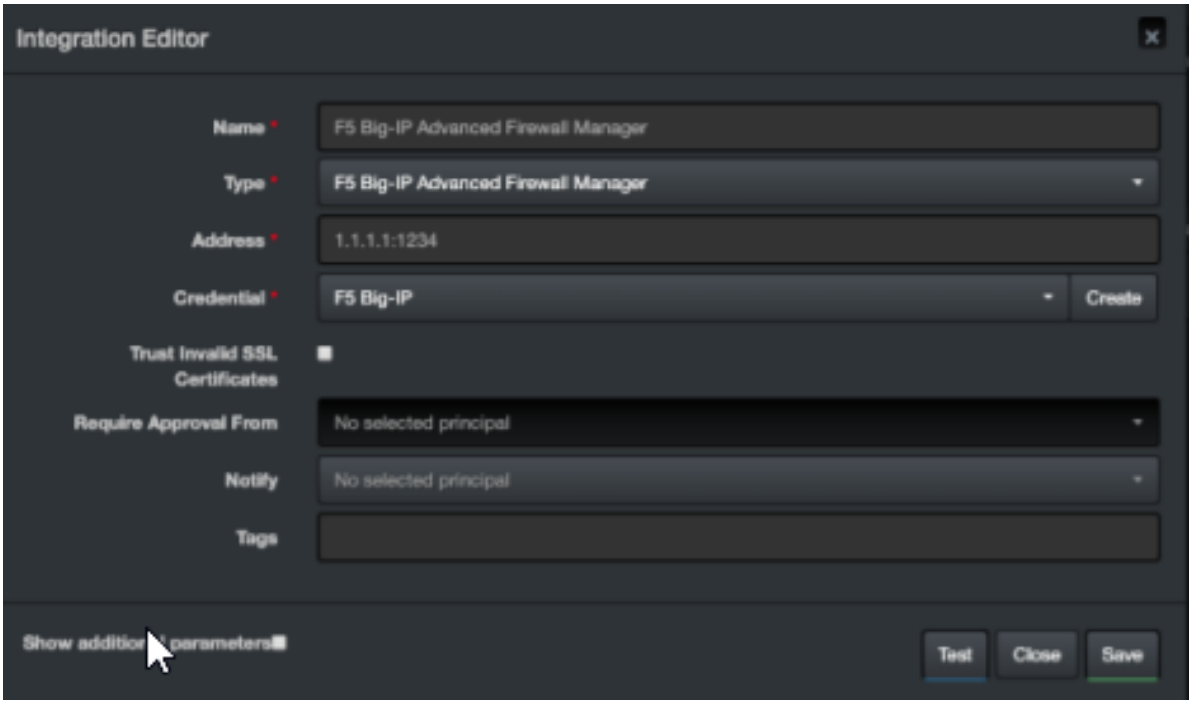
- Make sure SOAR has access to F5 Big-IP Advanced Firewall Manager integration's API as it connects to it using HTTPS.

Configuring SOAR

1. To create the integration, navigate to **Configuration** > **Integrations**.
2. Specify the following parameter values in the **Integrations Editor** form.

Parameter	Value
Name	Display name of integration
Type	F5 Big-IP Advanced Firewall Manager

Address	Address of the integration (in the format 1.1.1.1:1234 or abc.example.com:1234)
Credential	Credential that was defined for this integration under the **Credentials** menu
Trust Invalid SSL Certificates	Select this if device's certificate is self-signed or is not recognized by browsers
Require Approval From	Select users from the list who can provide approval before executing actions on this integration
Notify	Select user(s) from the list to notify when SOAR performs an action on this integration



3. Click **Test**. The following pop up will be displayed if your credential and address are valid.
4. Click **Save** to complete integration.

Integration Guide for FireEye HX

Integration Overview

FireEye HX is an endpoint threat detection and prevention solution. ArcSight SOAR integrates with FireEye HX through REST API to give enrichment and action capabilities to the users.

Integration Capabilities

Enrichment

- **IoC Scan:** HX can scan a given scope item in a target system and return information.
- **Detailed System Information:** HX can gather a target system information.
- **Script Execution:** HX supports different forensic data gathering scripts. These are XML formatted files that exist on HX installation. If customer wishes, they can import these script like files into Customization Library and then execute them through SOAR.

Action

Quarantine: HX quarantines a target system and reverts the quarantine if required.

Configuration

Configuring FireEye HX

- Make sure API services are enabled and create a api_admin user. To enable the services, see HX_API_v4+ guide.
- Access to the port number defined in the HX during installation as SOAR connects to FireEye HX.
- Define required access control rules if SOAR and FireEye HX are segregated.

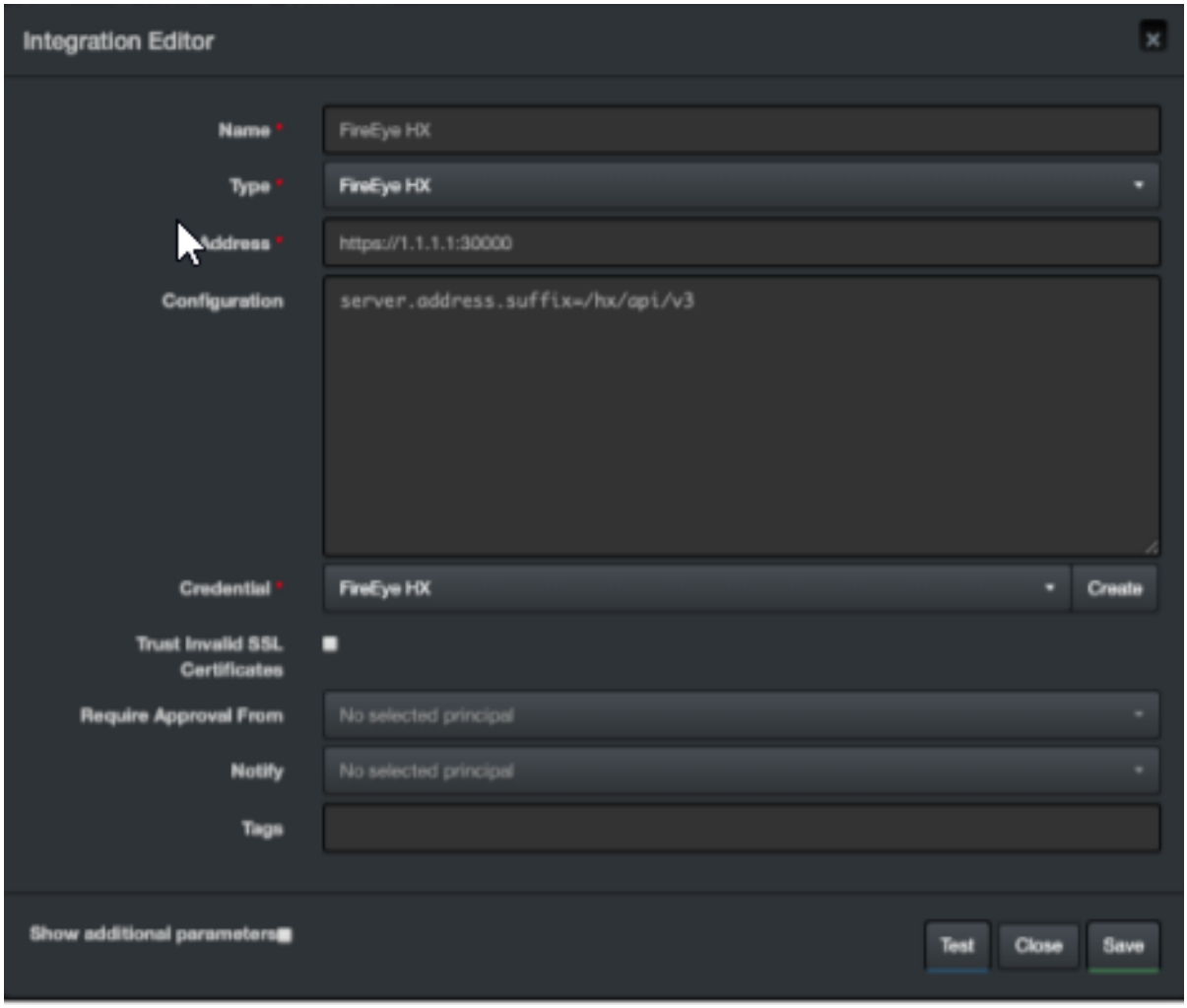
Configuring SOAR

SOAR configuration is standard and users need to specify **Name**, **Address** and **Credential fields**. Rest of the fields can be changed as required.

Note: Configuration field must not be changed by users.

1. To create the integration, navigate to **Configuration** > **Integrations**.
2. Specify the following parameter values in the **Integration Editor** form:

Parameter	Value
Name	Display name of the integration
Type	FireEye HX
Address	Address of the alert source (in the format http[s]://1.1.1.1:3000 or http[s]://abc.example.com:3000)
Configuration	Specify the following configuration parameter: `server.address.suffix=/hx/api/v3`
Credential	Credential defined under the **Credentials menu**
Trust Invalid SSL Certificates	Select this if device's certificate is self-signed or is not recognized by browsers
Require Approval From	Select users from the list who can provide approval before executing actions on this integration
Notify	Select users from the list to notify when SOAR performs an action on this integration



3. Click **Test** to test the integration.
4. Click **Save** to complete integration.

Integration Guide for Forcepoint Cloud Services

Integration Overview

SOAR works with Forcepoint Cloud Services to report uncategorized sites.

Integration Capabilities

Action

- Report

Configuration

Configuring Forcepoint Cloud Services

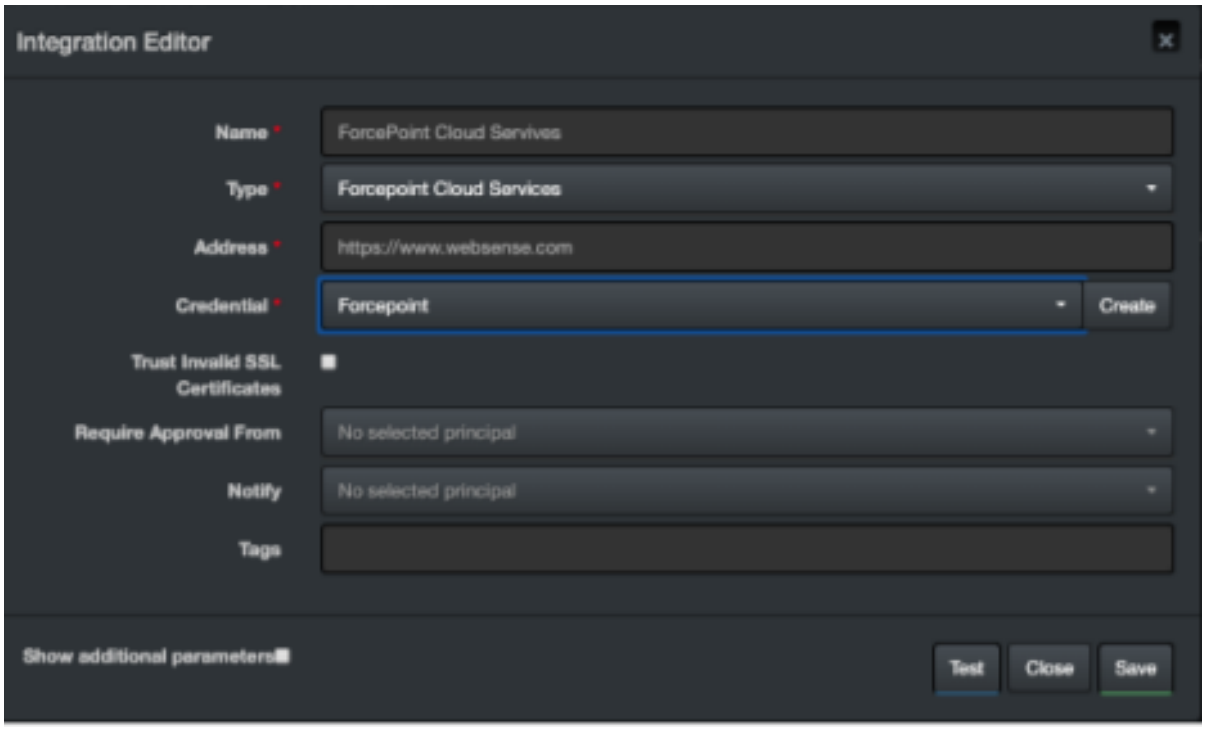
- Make sure SOAR has access to HTTPS as it connects to Forcepoint Cloud Services URL https://www.websense.com).
- A user account on Forcepoint/WebSense to use the **Sitelookup** tool.

Configuring SOAR

1. To create the integration, navigate to **Configuration** > **Integrations**.
2. Specify the following parameter values in the **Integrations Editor**.

Parameter	Value
Name	Display name of the integration
Type	Forcepoint Cloud Services
Address	Address of the integration (in the format http[s]://abc.example.com:3000)
Credential	Credential defined for this integration under the **Credentials** menu.

Trust Invalid SSL Certificates	Select this if device's certificate is self-signed or is not recognized by browsers
Require Approval From	Select users from the list who can provide approval before executing actions on this integration
Notify	Select users from the list to notify when SOAR performs an action on this integration



3. Click **Test** to test the integration.
4. Click **Save** to complete the integration.

Integration Guide for Forcepoint Content Gateway

Integration Overview

Forcepoint Web Content Gateway is a web proxy and cache that analyzes HTTP(S) requests in real-time and passes the traffic to Filtering Service for policy enforcement.

Integration Capabilities

ArcSight SOAR has the following integration capability with Forcepoint Web Content Gateway:

- Block Access to IP Addresses, URLs and Hostnames

Use Case: Blocking Phishing Domains

SOAR checks the inbox of user's email, for phishing reports and automatically creates an incident record on the service desk. During the investigation, SOAR extracts the malicious IP addresses, domains, and URLs in the message body and blocks access to Forcepoint Web Content Gateway. This can either be performed automatically within a playbook or manually by an analyst.

Also, SOAR uses threat intelligence (TI) feeds as an Alert Source and automatically blocks malicious domains/IP addresses reported by TI source on Forcepoint Web Content Gateway before any attack occurs.

Configuration

Prerequisites

- Current version of Forcepoint Web Content Gateway.
- Access to HTTPS as SOAR connects to Forcepoint Web Content Gateway Policy API
- Access to 15873/tcp port

Configuring Facepoint Web Content Gateway

1. Forcepoint Management API does not get installed by default. To complete the integration, install this service on the server or appliance. Also, the configuration steps differ with the usage of the server. For the complete instructions, see Management API Installation Guide.
2. After installing Management API components, use the Forcepoint Security Manager to configure the account used for authentication. To enable the communication, see ***Enabling communication between Management API clients and servers*** in the Management API Installation Guide.

Configuring SOAR

1. Click **Configurtion** > **Credentials** > **Create Credential**.
2. Specify the following parameter values in the **Credential Editor** form:
   a. **Internal credential:**

Parameter	Value
Type	Internal credential
Name	Display name of the credential set (For example, Forcepoint WCG Credentials)
Username	Username configured on Forcepoint Management API
Password	Password for the user configured on Forcepoint Management API.
Private Key	Empty

   b. **Credential Store:**

Parameter	Value
Type	External credential
Name	Name of the credential with pull path of the safe on store.

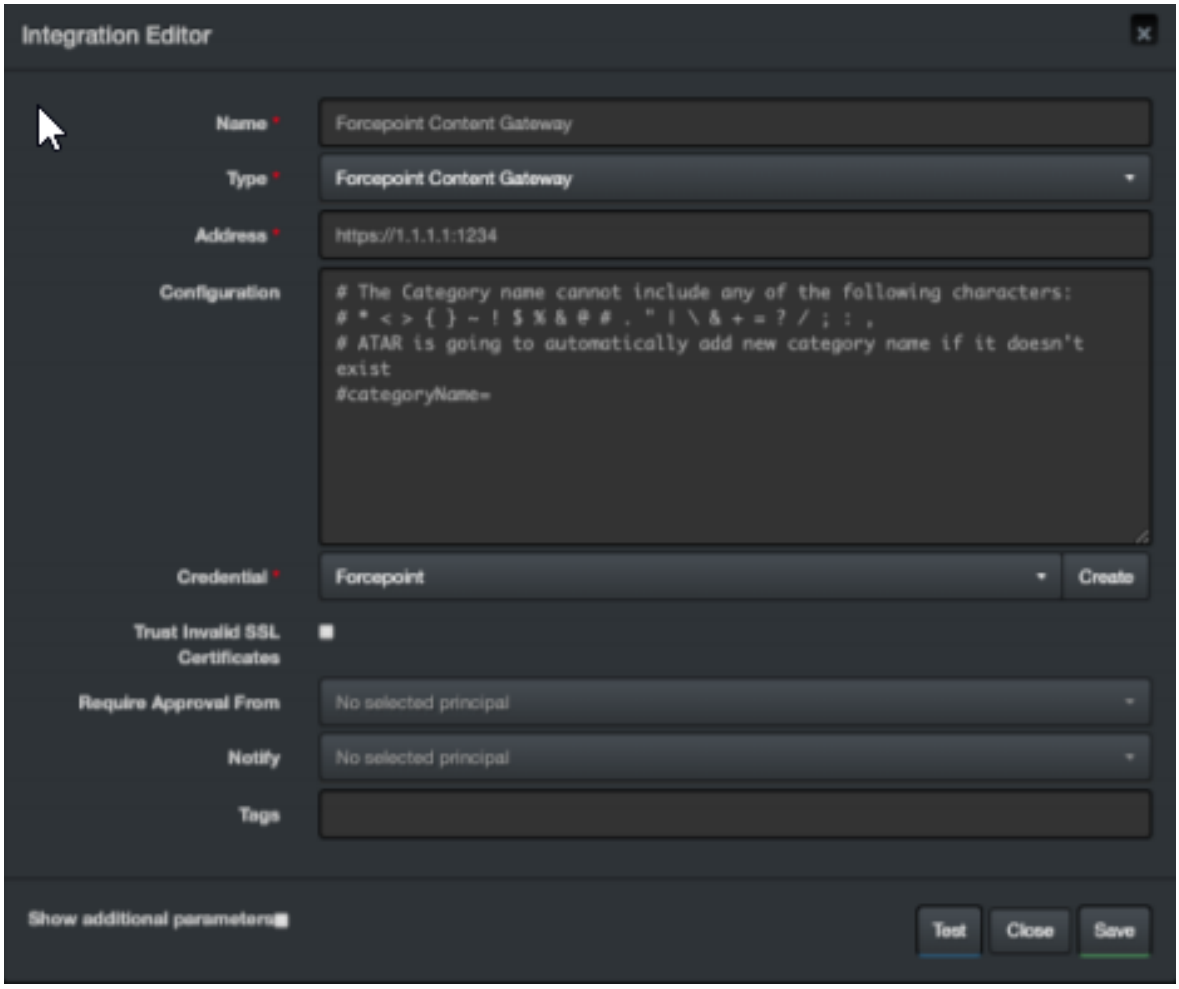
3. Click **Configuration** > **integrations** > **Create Integration.**
4. Specify the following configuration parameter values in the **Configuration form**:

Parameter	Value
Name	Display name of Forcepoint Web Content Gateway integration on SOAR
Type	Forcepoint Web Content Gateway

Address	Address of the integration (in the format https://192.168.2.99:15:15873).

Configuration	Specify the following configuration parameters:
	`# The Category name cannot include any of the following characters:`
	`# * < > { } ~ ! $ % & @ # . " | \ & + = ? / ; : ,`
	`# SOAR is going to automatically add new category name if it doesn't exist`
	`categoryName=SOAR_BLOCK`

Credential	Name of the credential set created on step 2. (For example, Forcepoint WCG Credentials)
Trust Invalid SSL Cerificates	Select this if Engine's certificate is self-signed or is not recognized by browsers.
Require Approval From	Select users from the list who can provide approval before executing actions on this integration.
Notify	Select users from the list to notify when SOAR performs an action on this integration



5. Click **Test**. The following pop up will be displayed if your credentials and address are valid.
6. Click **Save** to complete integration.

Additional Notes

- The **categoryName** you provide in the Configuration section is API-Managed but not managed by UI. If the category does not exist on the device, SOAR creates it automatically.

Integration Guide for ForeScout CounterACT NAC

Integration Overview

ForeScout CounterACT NAC provides virtual insight into any device connected across the enterprise and gives a single-pane-of-glass perspective. ForeScout discovers devices in real-time, then classifies, assesses, and monitors these devices. Also, this platform provides agent-less control and continuous monitoring across heterogeneous environments. Enables to trigger actions to notify, monitor, and remediation.

Integration Capabilities

SOAR has the following integration capability with ForeScout CounterACT NAC:

Action Capabilities

- Assign Policy to Host

Enrichment Capabilities

- Host information query by Network Address
- Host information query by Username
- Host information query by MAC Address
- Host information query by Computer Name

Use Case: Isolating Mal-behaving PC

SOAR integrates with ForeScout CounterACT NAC, while responding to an incident it applies a policy to mal-behaving computers and prevents further spread of the attack. A policy to the host can either be applied automatically within a playbook or manually by an analyst.

Configuration

Prerequisites

- Current version of ForeScout CounterACT NAC
- Access to SSH protocol(22/tcp port) as SOAR connects to ForeScout CounterACT NAC using SSH protocol.
- Access to 443/tcp port as enrichment plugin connects to ForeScout CounterACT NAC server
- A shell user account needs to be created for SOAR to connect to ForeScout

CounterACT NAC

Configuring ForeScout CounterACT NAC

1. Login to ForeScout CounterACT NAC appliance.
2. Create a shell account by executing the following command in the command prompt:

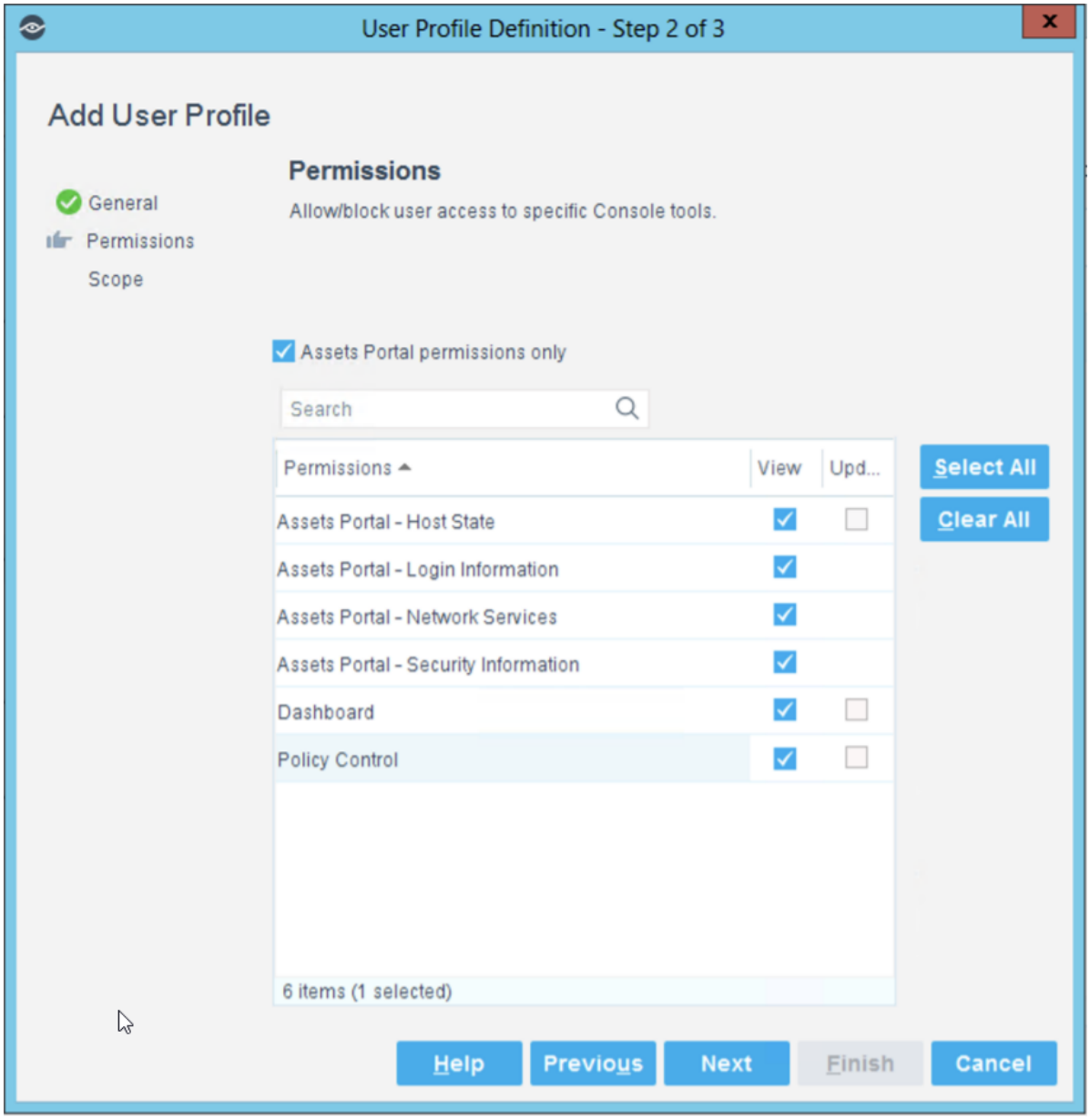
   ```
   $ useradd -s /bin/bash -m -d /home/soar soar
   ```

   ```
   $ passwd atar
   ```

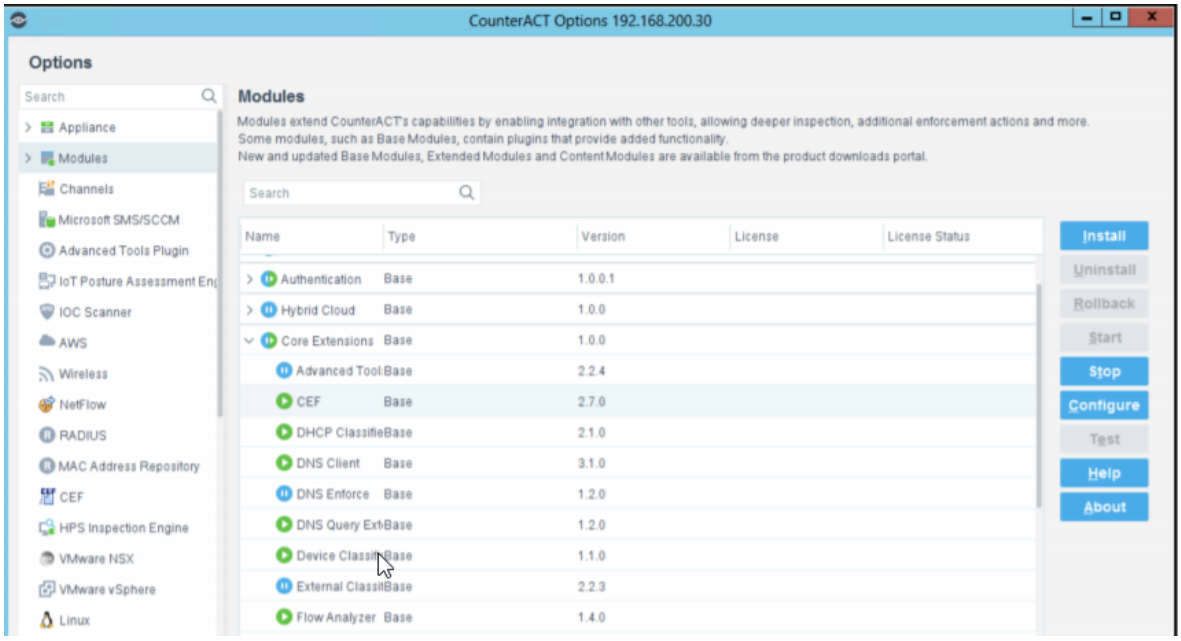
3. To allow new user to execute fstool command without the need to enter the password, add the following line to sudo configuration (/etc/sudoers)

   ```
   soar ALL=(root) NOPASSWD: /usr/local/forescout/bin/fstool
   ```

4. To use enrichment capabilities, add or use an existing web management user with the following permission:



5. Login to Forescout **Management Interface.**
6. Enable **CEF service**.



7. Navigate to **Policy** and edit one of the existing policies or create a new one.
8. To edit condition of a rule, add “SIEM Message” as Criteria and select desired action.

Note: Make a note or save the SIEM message to use while configuring SOAR.

Configuring SOAR

1. Click **Configuration** > **Credentials** > **Create Credential.**
2. Specify the following parameter values in the **Credential Editor** form:
   a. **Internal Credential**

Parameter	Value
Type	Internal Credential
Name	Display name of credential set (For example, ForeScout CounterACT NAC Credential)
Username	Username created for SOAR on ForeScout CounterACT NAC
Password	Password of the user that was created for SOAR on ForeScout CounterACT NAC
Private Key	Empty

   b. **Internal Credential**

Parameter	Value
Type	Internal Credential
Name	Display name of credential set (For example, ForeScout CounterACT NAC Credential)
Username	Username created for SOAR on ForeScout CounterACT NAC for web management user (2.2.3).
Password	Password of the user you have created for SOAR on ForeScout for web management user (2.2.3).
Private Key	Empty

Note: Make a note or save the credential ID to use it in device configuration (2.3.4).

c. **Credential Store:**

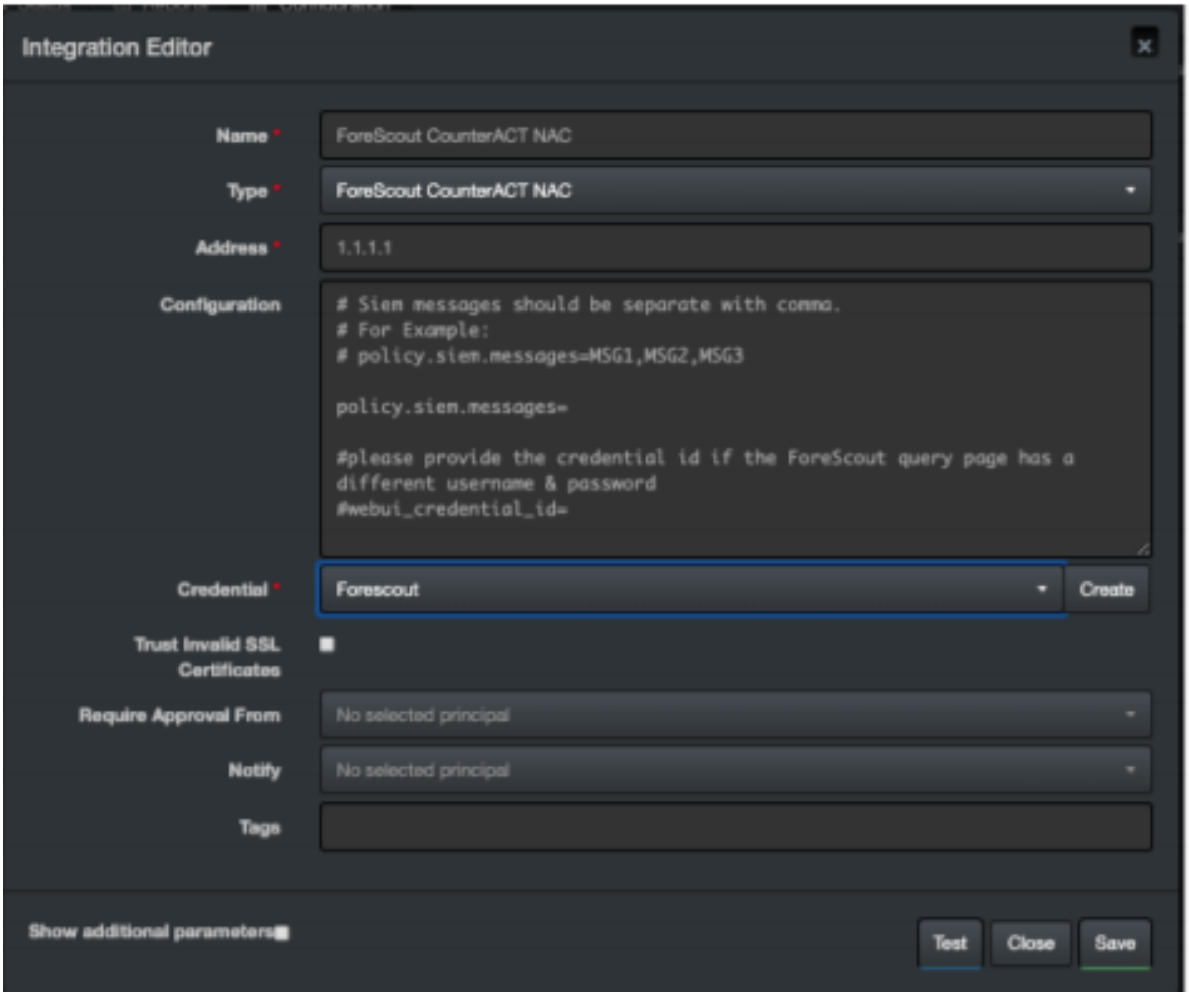
Parameter	Value
Type	External Credential
Name	Name of the credential with pull path of the safe on store.

3. Click **Confiugration** > **Integrations** > **Create Integration.**
4. Specify the following parameter values in the **Configuraiton** form:

Parameter	Value
Name	Display name of Database Server integration on SOAR
Type	ForeScout CounterACT NAC
Address	Address of the integration (in the format 192.168.1.1)

Configuration	Specify the following configuration parameters.
	`# Supported versions are: v1 (for version 8.0) and v2 (for version 8.1.3). Default version is v1`
	`#version=`
	`# Siem messages should be separate with comma.`
	`# For Example:`
	`# policy.siem.messages=MSG1,MSG2,MSG3`
	`policy.siem.messages=`
	`# please provide the credential id if the ForeScout query page has a`
	`# different username & password`
	`webui_credential_id=(Credential id that you made a note in step 2.3.4)`

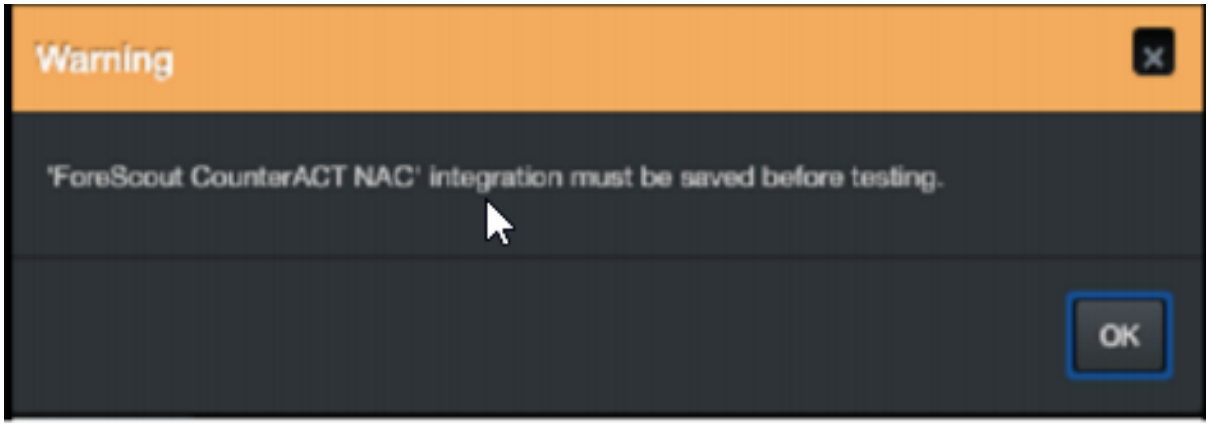
Credential	Name of the credential set created on step 2. (For example, ForeScout CounterACT NAC Credential)
Trust Invalid SSL Certificates	Select this if device's certificate is self-signed or is not recognized by browsers
Require Approval from	Select users from the list who can provide approval before executing actions on this integration
Notify	Select users from the list to notify when SOAR performs an action on this integration.



5. Click **Test**. The following pop up will be displayed if your credential and address are valid.
6. Click **Save** to complete integration.

Additional Notes

- ForeScout CounterACT NAC integration is an Advanced Script, and the content of the default script is accessible under **Configuration** > **Customization** Library.
- While defining the integration for the first time, you might encounter the following warning message, which is the expected behavior for this type of integration.


Warning
'ForeScout CounterACT NAC' integration must be saved before testing.
OK

Integration Guide for Fortinet FortiGate Firewall

Integration Overview

ArcSight SOAR uses Fortinet FortiGate Firewall to block IP addresses on the network perimeter and terminates sessions using the incident scope.

Integration Capabilities

- Action
- Block
- Disconnect
- Custom Script

Configuration

Configuring FortiGate Firewall

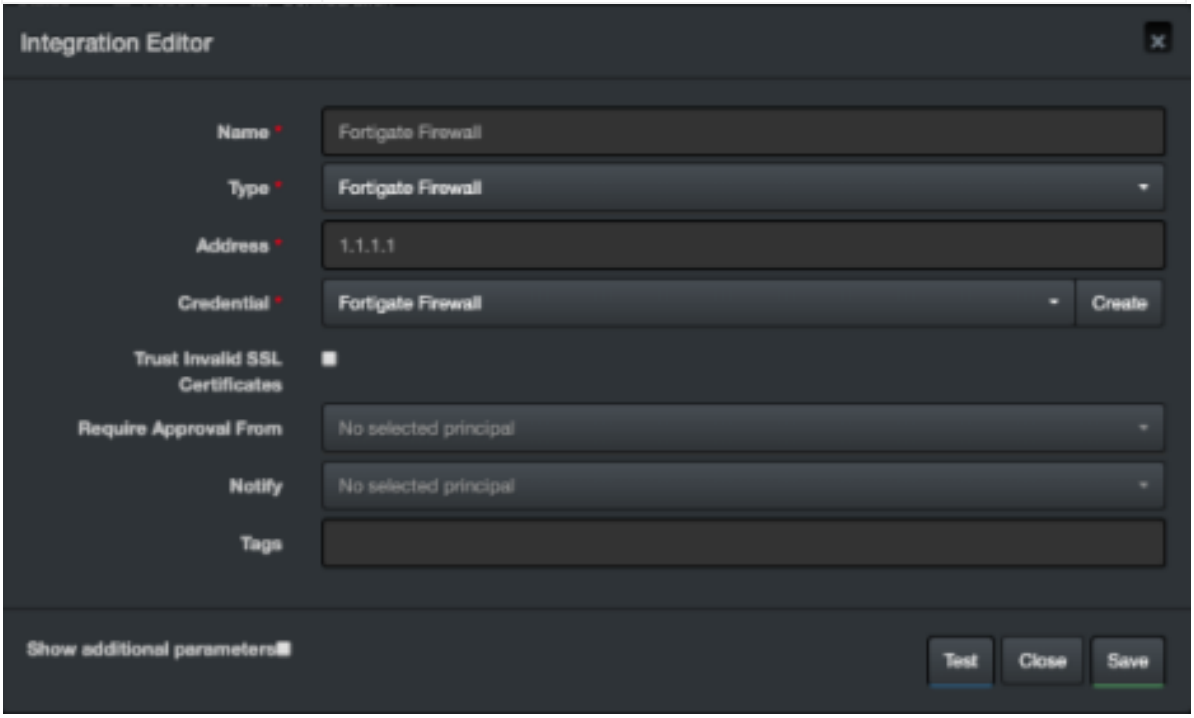
- Make sure SOAR has access to SSH as it connects to FortiGate Firewall integration using it
- A user's credential with admin role
- An empty rule to be used by SOAR to block offending IP addresses

Configuring SOAR

1. To create the integration, navigate to **Configuration** > **Integrations**.
2. Specify the following parameter values in the **Integrations editor**:

Parameter	Value
Name	Display name of the integration
Type	Fortigate Firewall
Address	Address of the integration (in the following format: 1.1.1.1 or abc.example.com)
Credential	Credential defined under the **Credentials menu**

Trust Invalid SSL Certificates	Select this if device's certificate is self-signed or is not recognized by browsers
Require Approval From	Select users from the list who can provide approval before executing actions on this integration
Notify	Select users from the list to notify when SOAR performs an action on this integration



3. Click **Test** to test the integration.
4. Click **Save** to complete the integration.

Additional Notes

You might have to review the actions that are defined and executed using the Fortigate Firewall custom scripts for SOAR. To access these custom scripts, navigate to **Configuration** > **Custom Scripts**.

The following custom scripts are specific to this device:

- FortiGate Firewall SSH Device Action (Block) Default Template
- FortiGate Firewall Availability Check Default Template

Integration Guide for Fortinet FortiAnalyzer

Integration Overview

Fortinet FortiAnalyzer is a central log collection and analysis tool for Fortinet products. SOAR can query FortiAnalyzer (FAZ) for scope items to enrich incident data and to search the past events for emerging threats.

Integration Capabilities

ArcSight SOAR has the following enrichment capabilities with Fortinet FortiAnalyzer:

- Accepted Traffic Logs : This query returns accepted traffic logs to or from the selected scope item and the time frame might be between 1 hour to 12 hours.
- URL Access Logs : This query returns the events that record access to the selected URL and the time frame might be between 1 hour to 12 hours.

Configuring Fortinet FortiAnalyzer

Web services must be enabled on the network interface to which the client connects.

1. To enable web services for an interface, navigate to **System Settings** > **Network** > **Interface**.
2. Select **Edit** for the interface for which you need to enable the web services.
3. In the **Administrative Access** section, select **Web Service**.
4. Select **OK** to apply the changes.
5. Create a user with a custom profile.

Note: This user profile requires access to **Log View/FortiView/NOC - SOC** component and **ADOM's** SOAR.

Configuring SOAR

1. Click **Configurtion** > **Credentials** > **Create Credential**.
2. Specify the following parameter values in the **Credential Editor** form:

a. **Internal credential:**

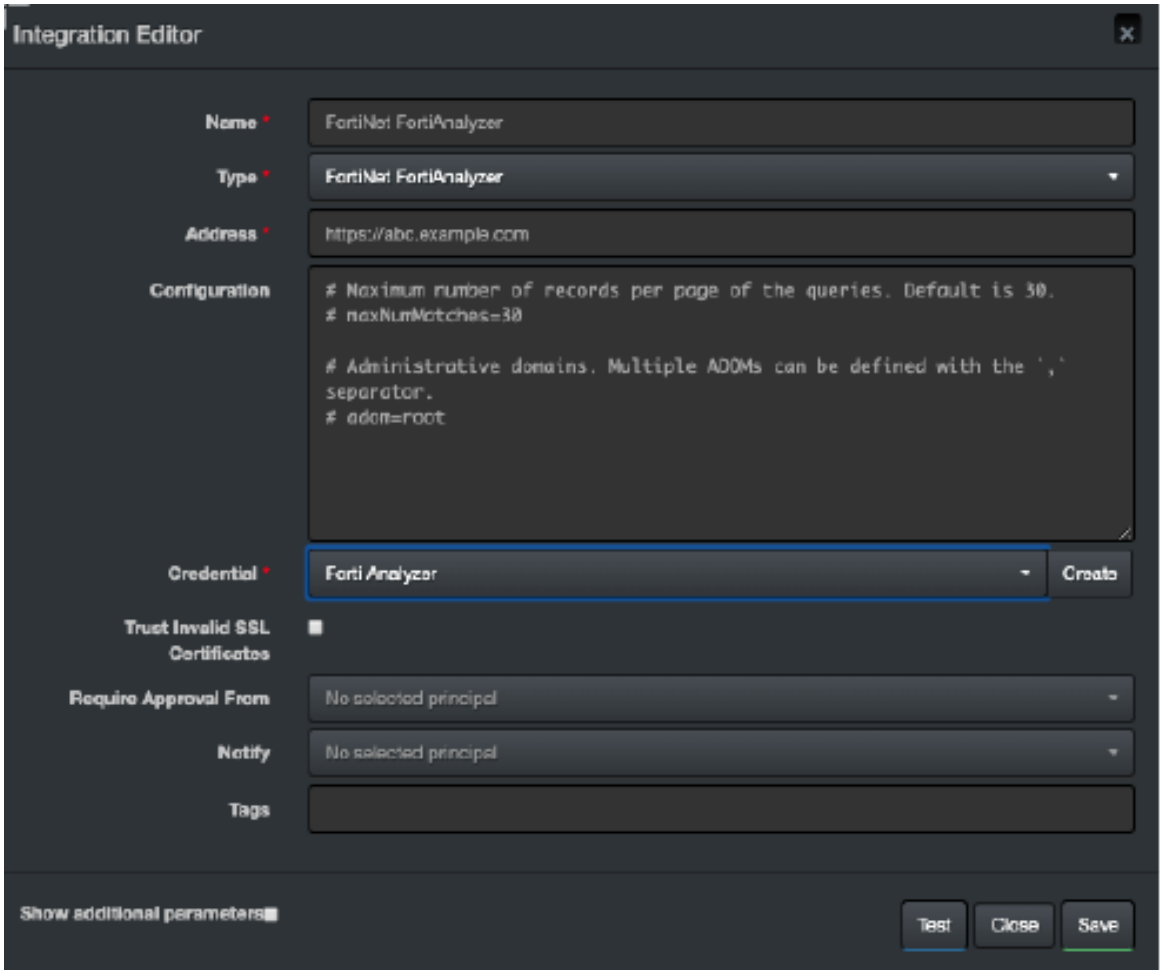
Parameter	Value
Type	Internal credential
Name	Display name of the credential set (For example, Fortinet FortiAnalyzer)
Username	API Key created on Fortinet FortiAnalyzer
Password	API Password for the key created on Fortinet FortiAnalyzer
Private Key	Empty

b. **Credential Store:**

Parameter	Value
Type	External credential
Name	Name of the credential with pull path of the safe on store

3. Click **Configuration** > **integrations** > **Create Integration**.
4. Specify the following configuration parameter values in the **Configuration form**:

Parameter	Value
Name	Display name of Fortinet FortiAnalyzer integration on SOAR
Type	Fortinet FortiAnalyzer
Address	Address of the integration (in the following format: 1.1.1.1 or http[s]://abc.example.com)
Credential	Name of the credential set created on step 2 (for example, Fortinet FortiAnalyzer Credentials)
Configuration	Specify the following configuration parameters: `maxNumMatches`: Define the number of results SOAR shows per page of query `adom`: ADOM's SOAR query to get logs from
Require Approval From	Select users from the list who can provide approval before executing actions on this integration
Notify	Select users from the list to notify when SOAR performs an action on this integration



5. Click **Test** to test the integration.
6. Click **Save** to save the integration.

Integration Guide for Fortinet FortiDDoS

Integration Overview

FortiDDoS is a network behavior anomaly (NBA) prevention system that detects and blocks attacks that intend to disrupt network service (distributed denial of service (DDoS) attacks) by over utilizing server resources.

Integration Capabilities

ArcSight SOAR has the following integration capabilities with FortiDDoS:

- Block IP and Hostname

Use Case: Blocking malicious IP on peripheral

SOAR integrates with FortiDDoS to block malicious IP addresses detected while responding tp an incident. Blocking can be performed automatically within a playbook or manually by an analyst.

Configuration

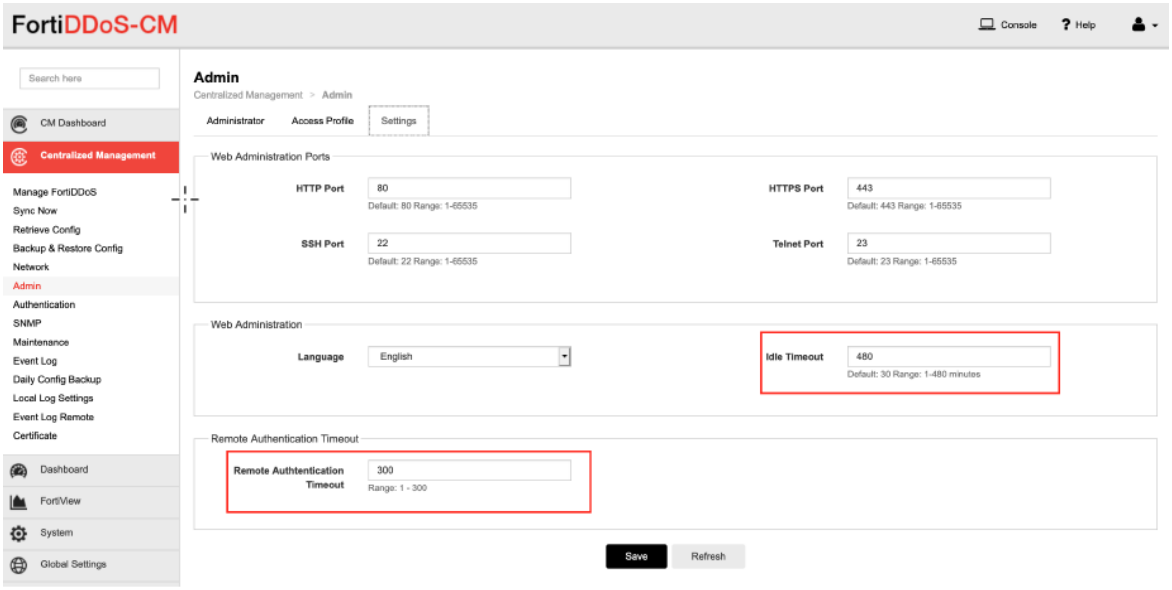
Prerequisites

- FortiDDoS version 4.7 and 5.1
- Access to tcp port 443 as SOAR connects to FortiDDoS' API using HTTPS
- An administrator user account for SOAR to connect to FortiDDoS

Configuring FortiDDoS

1. To add a new SOAR user with the required access profile permissions, navigate to **System** > **Admin** > **Access Profile**.
2. In the Access profile form, select **Global Settings** and **Protection profiles**with **Read & Write** permissions.
3. Navigate to **System** > **Admin** > **Administrator**.
4. To add an administrator with the profile created in the previous step, select **Enable** for **Allow API Access**.

5. (Optional) To specify **Remote Authentication** and **Idle timeout** values, navigate to **Centralized Management** > **Admin**.



6. Click **Save** to save the changes.

Configuring SOAR

1. Click **Configuration** > **Credentials** > **Create Credential**.
2. Specify the **Credential Editor** with the following parameter values:
   a. **Internal Credential**

Parameter	Value
Type	Internal Credential
Name	Display name of credential set (For example, FortiDDoS Credentials)
Username	User created on FortiMail for SOAR
Password	Password of the user that was created for SOAR on FortiMail
Private Key	Empty

   b. **Credential Store:**

Parameter	Value
Type	External Credential
Name	Name of the credential with pull path of the safe on store

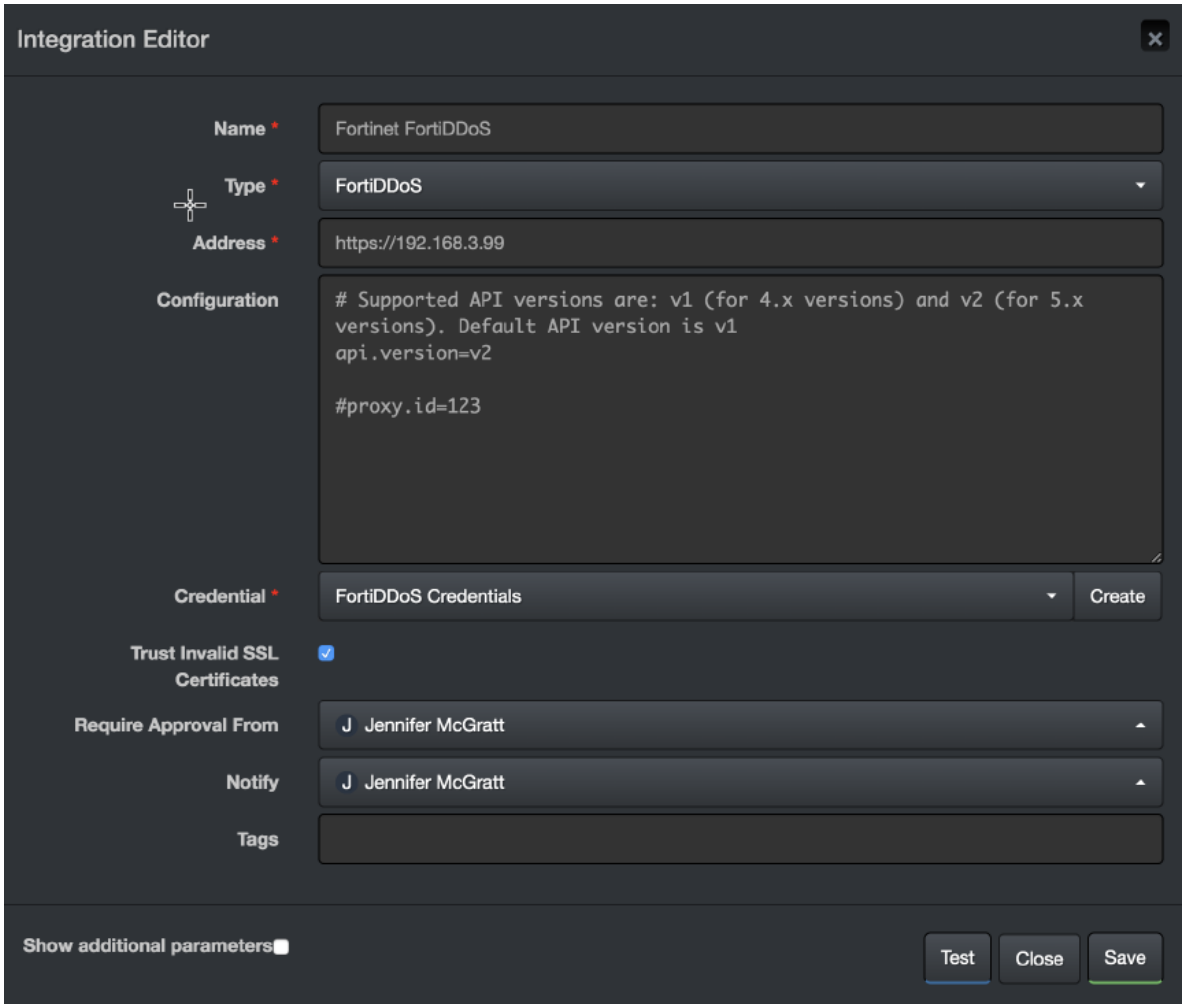
3. Click **Configuration** > **Integrations** > **Create Integration**.

4. Specify the following parameter values in the **Configuration** form:

Parameter	Value
Name	Display name of FortiDDoS integration on SOAR
Type	FortiDDoS
Address	Address of the integration (in the following format: https://192.168.3.99)

Configuration	Specify the following configuration parameters:
	`# Supported API versions are: v1 (for 4.x versions)`
	`and v2 (for 5.x versions). Default api.version=v2`
	`#proxy.id=123`

Credential	Name of the credential set created on step 2 (For example, FortiDDoS Credentials)
Trust Invalid SSL Certificates	Select this if Integrations's certificate is self-signed or is not recognized by browsers.
Require Approval From	Select users from the list who can provide approval before executing actions on this integration
Notify	Select users from the list to notify when SOAR performs an action on this integration



5. Click **Test** to test the integration.
6. Click **Save** to complete the integration.

Integration Guide for Fortinet FortiGate API

Integration Overview

Fortinet FortiGate is an industry leading next generation security platform.

Integration Capabilities

ArcSight SOAR has the following integration capabilities with Fortinet FortiGate API:

- Action
- Block IP
- Block FQDN
- Block URL

Use Case: Blocking malicious artifacts detected through alerts

SOAR automatically executes playbooks and blocks malicious artifacts on FortiGate platform. The artifacts IP, Domain and URL can be blocked using SOAR.

Configuration

Prerequisites

- Access to tcp port 443 as SOAR connects to Fortinet FortiGate API using HTTPS
- A user account with necessary permissions on the FortiGate platform

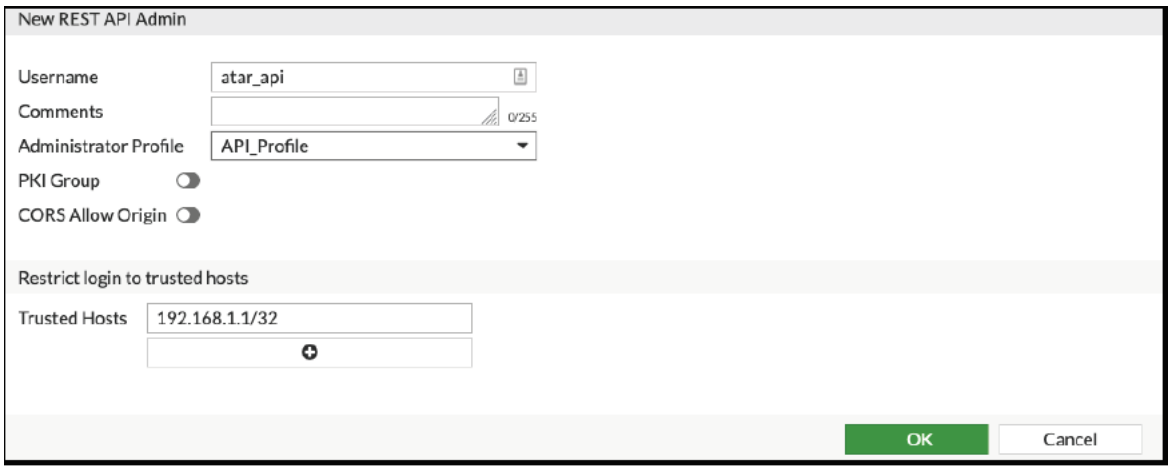
Configuring Fortinet FortiGate

1. To create a user, navigate to **System** > **Administrators**.
2. Click **Create New** and select **REST API Admin**.
3. Specify the following values in the **New REST API Admin** form:

   Username: <SOAR user name>

   Administrator Profile: <profile name>

   Trusted Hosts: A subnet that covers SOAR's API address

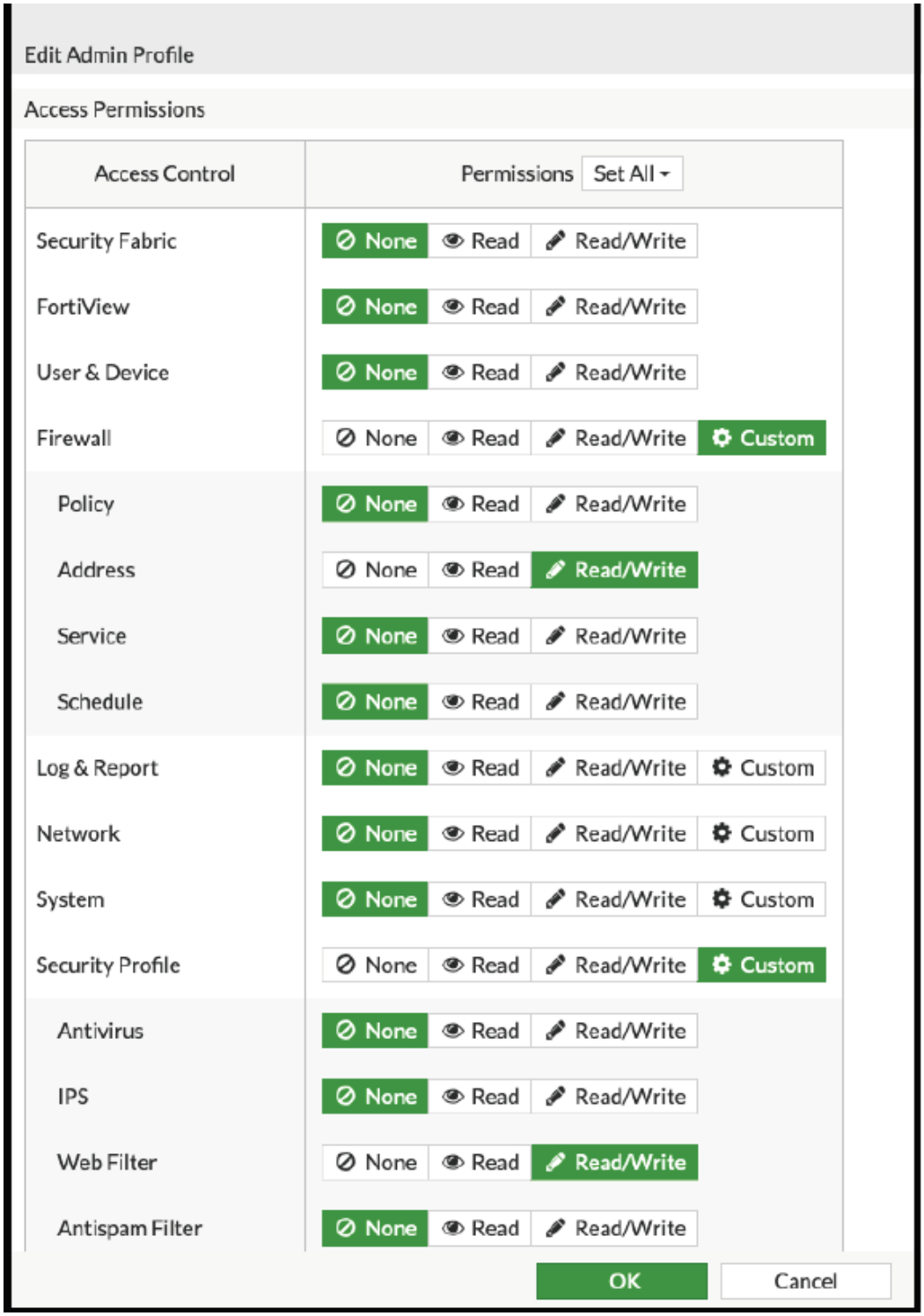


Note: Use the IP address that SOAR uses and **0.0.0.0/0** must not be used as an IP address.

4. To create a profile, click **+** in the **Admin Profile** window.
5. Select **Read/Write** persmissions for the following groups:

   Firewall > **Address**

   Security > **Web Filter**



6. Click **OK** to save the profile and save the API key.

Configuring SOAR

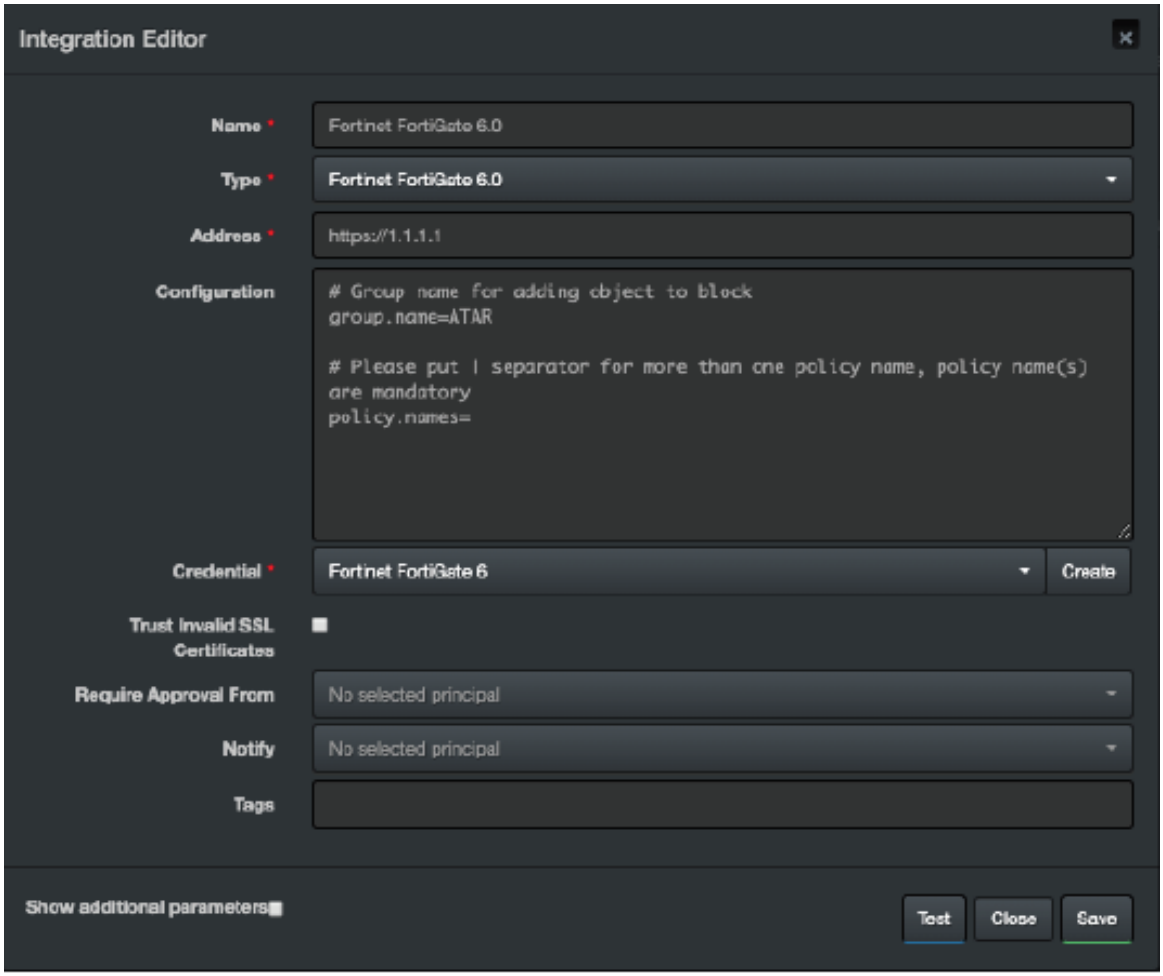
1. Click **Configuration** > **Credentials** > **Create Credential**.
2. Specify the **Credential Editor** with the following parameter values:
   a. **Internal Credential**

Parameter	Value
Type	Internal Credential
Name	Display name of credential set (For example, Fortinet FortiGate Credentials)
Username	Empty
Password	Empty
Private Key	Enter the API Key generated by FortiGate

Note: Fortinet FortiGate requires private key and External Credential is not used.

3. Click **Configuration** > **Integrations** > **Create Integration**.
4. Specify the following parameter values in the **Configuration** form:

Parameter	Value
Name	Display name of Fortinet FortiGate integration on SOAR
Type	Fortinet FortiGate 6.0
Address	Address of the firewall
Configuration	Specify the following configuration parameters: `group.name`: Group name for adding objects to be blocked. This Address Group will be created on FortiGate and then can be used in policies as the admin see fit `policy.names`: Policy names to be used to block URL. '|' is used as separator for policies and SOAR writes the URL to all the policies defined
Credential	Name of the credential set that was created on step 2 (For example, Fortinet FortiGate Credentials)
Trust Invalid SSL Certificates	Select this if Integrations's certificate is self-signed or is not recognized by browsers
Require Approval From	Select users from the list who can provide approval before executing actions on this integration



5. Click **Save** to complete the integration.

Additional Notes

- The API Key to work properly requires access to HTTPS and for security reasons as well.

  Note: By default, HTTP access is enabled in FortiGate. However, in production environment, it is recommneded to disable the HTTP access.

- If you have multiple policies on the integration configuration and if one of the policy's URL filter is disabled, SOAR with Fortinet integration displays no specific error message. In such case, you might encounter the following error message:

  None of policy names in t he configuration are present in the Fortinet FortiGate server.

Integration Guide for Fortinet FortiMail

Integration Overview

Fortinet FortiMail secure email gateway utilizes the latest technologies and security services from FortiGuard Labs to protect from common and advanced threats while integrating robust data protection capabilities to avoid data loss.

Integration Capabilities

ArcSight SOAR has the following integration capabilities with FortiMail:

- Add to Block List
- Block

Use Case: Blocking malicious sender

SOAR integrates with FortiMail to block malicious email addresses detected while responding to an incident. The blocking can either be performed automatically within a playbook or manually by an analyst.

Configuration

Prerequisites

- FortiMail version 6.2.2(GA) and later
- Access to tcp port 443 as SOAR connects to FortiMail API using it
- An administrator user account for SOAR to connect to FortiMail

Configuring FortiMail

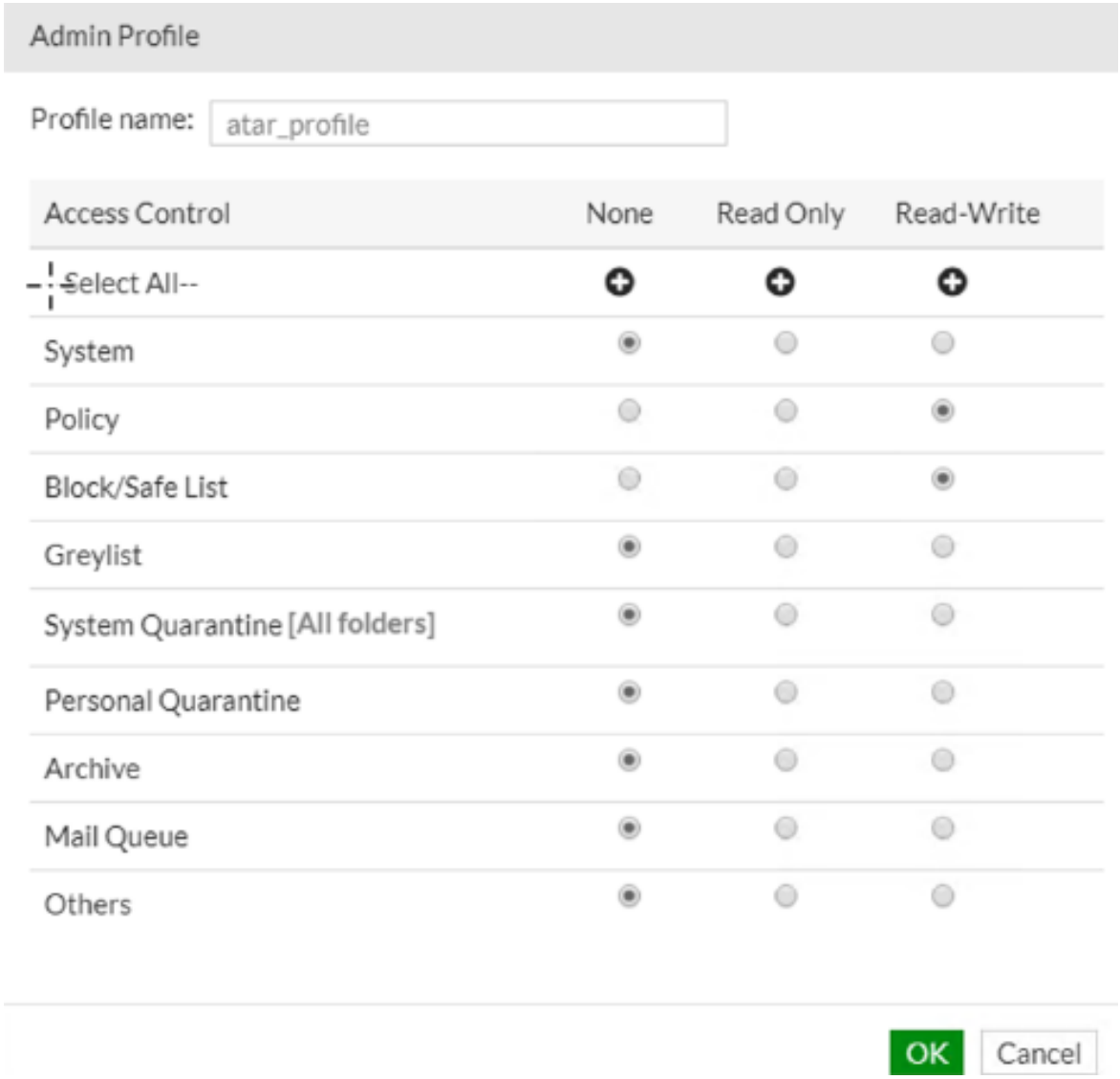
1. By default, REST-API service is disabled on FortiMail. To enable it, use the following CLI command:

```
config system global
```

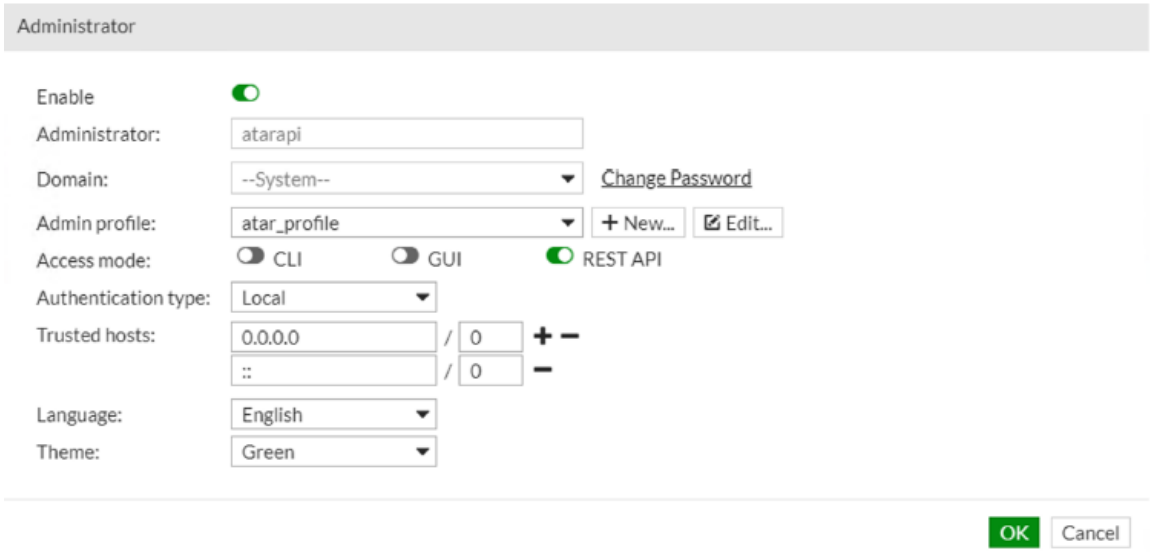
```
set rest-api enable
```

```
end
```

2. Navigate to **System** > **Administrator** > **Admin Profile**.
3. Select **Policy**, **Block/Safe List** with **Read-Write** support and create an admin profile in the **Admin Profile** form.



4. Navigate to **System** > **Administrator** > **Administrator**.
5. Create a new administrator account with the profile that you have created in the previous step.



Configuring SOAR

1. Click **Configuration** > **Credentials** > **Create Credential**.
2. Specify the **Credential Editor** with the following parameter values:
   a. **Internal Credential**

Parameter	Value
Type	Internal Credential
Name	Display name of credential set (For example, FortiMail Credentials)
Username	User created on FortiMail for SOAR
Password	Password of the user created on FortiMail for SOAR
Private Key	Empty

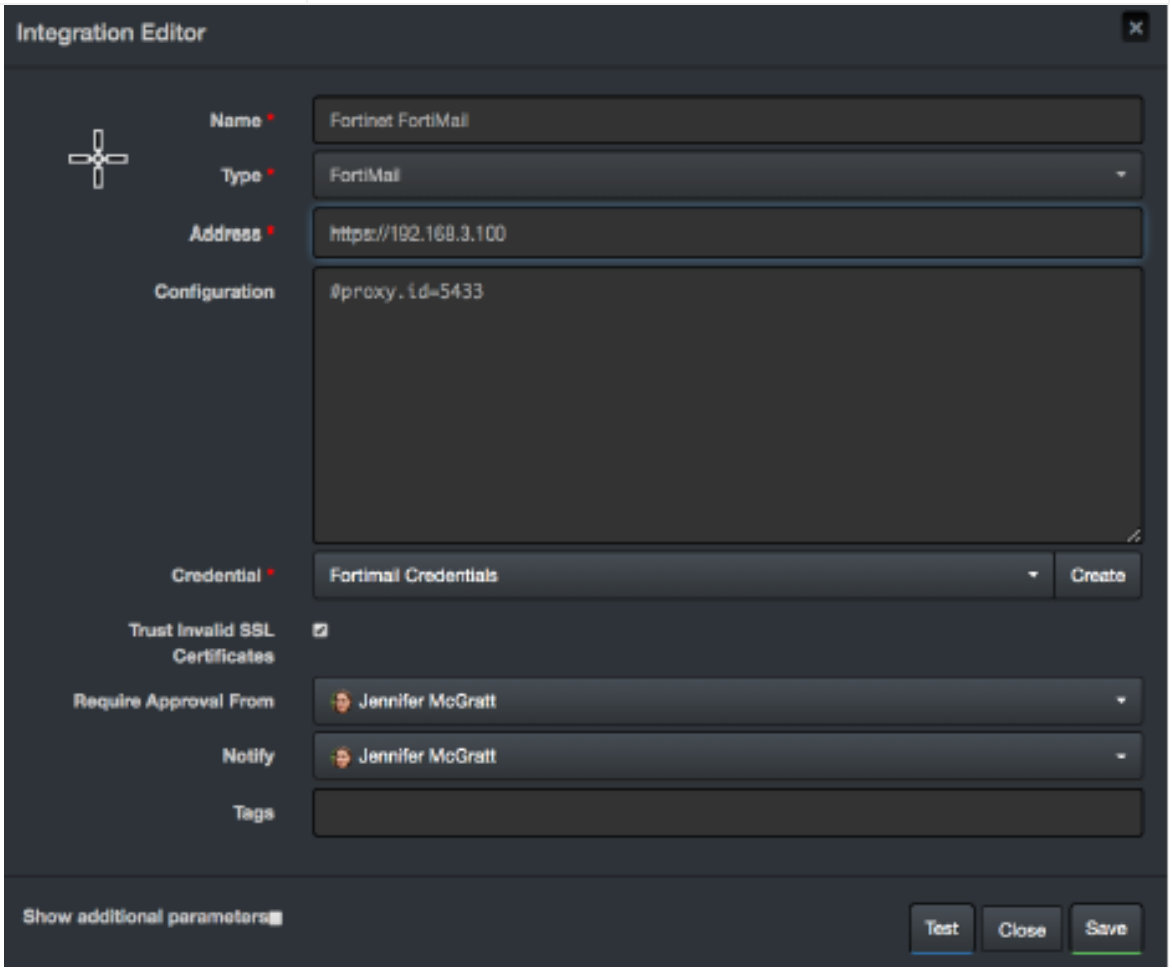
   b. **Credential Store:**

Parameter	Value
Type	External Credential
Name	Name of the credential with pull path of the safe on store

3. Click **Configuration** > **Integrations** > **Create Integration**.
4. Specify the following parameter values in the **Configuration** form:

Parameter	Value
Name	Display name of FortiMail integration on SOAR

Type	FortiMail
Address	Address of the integration (in the following format: https://192.168.3.100)
Configuration	Specify the following configuration parameters: `#proxy.id=5433`
Credential	Name of the credential set created on step 2 (For example, FortiMail Credentials)
Trust Invalid SSL Certificates	Select this if Integrations's certificate is self-signed or is not recognized by browsers.
Require Approval From	Select users from the list who can provide approval before executing actions on this integration
Notify	Select users from the list to notify when SOAR performs an action on this integration



5. Click **Test** to test the integration.
6. Click **Save** to complete the integration.

Additional Notes

Add to Block List capability uses the **Security** > **System** > **Blocklist**, whereas **Block capability**

uses the **Policy** > **Access Control**.

Integration Guide for Fortinet FortiManager

Integration Overview

Fortinet FortiManager is a centralized management unit for Fortinet family devices. It provides best compliance practices and workflow automation.

Integration Capabilities

ArcSight SOAR has the following integration capabilities with FortiManager:

- Block file on an connected Fortinet family device (For example, Fortinet NGFW, Fortinet FortiMail, etc)
- Block IP address on an connected Fortinet family device (For example, Fortinet NGFW, Fortinet FortiMail, etc)
- Block username on an connected Fortinet family device (For example, Fortinet NGFW)
- Block email on an connected Fortinet family device (For example, Fortinet FortiMail)

Use case: Mitigating Compromised Account Cases

SIEM, with the help of intelligence sources, creates an alarm. It compromises the suspected email accounts of the employees. SOAR integrates with Fortinet FortiManager and automatically blocks the outgoing emails and the incoming and outgoing traffic. This blocking can either be performed automatically within a playbook or manually by an analyst.

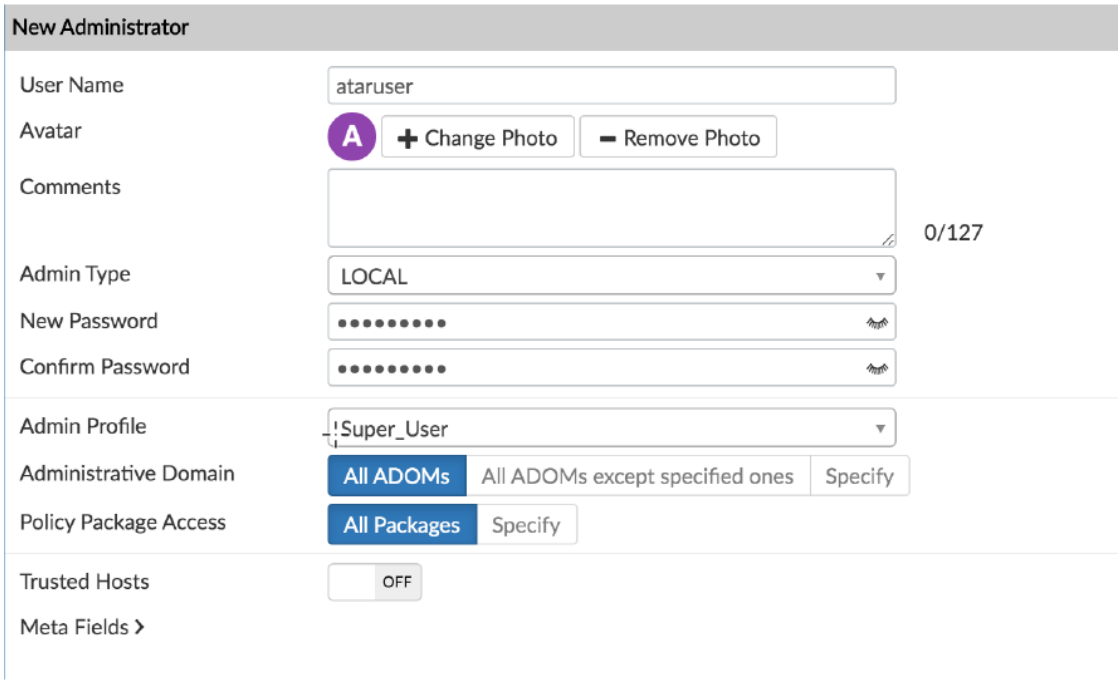
Prerequisites

- Fortinet FortiManager v5.6.2-build1631 180124 (GA) firmware version as SOAR supports it
- Access to tcp port 443 as SOAR connects to Fortinet FortiManager using HTTPS
- A user account for SOAR to connect to Forti Manager

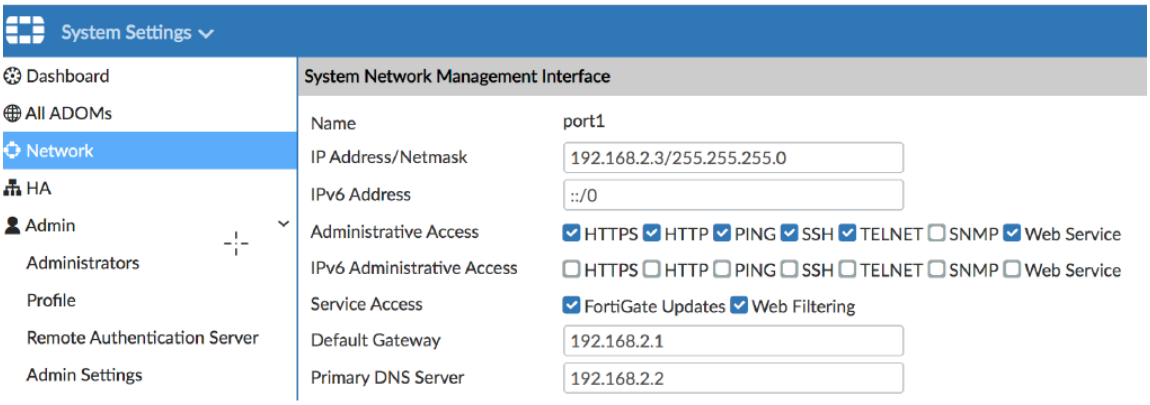
Configuration

Configuring FortiManager

1. Navigate to **System Settings** > **Admin** > **Administrators**.
2. To create a profile with Super_User account, specify the following values in the **New Administrator** form:
   - **Username:** <SOAR username>
   - **Admin Type:** Local
   - **New Password:** <Specify the password>
   - **Confirm Password:**< Confirm the password entered in the **Password** field>
   - **Admin Profile:** Super_User



3. Navigate to **System Settings** > **Network**.
4. Enable the **Web Service** in the **Administrative Access**.



Configuring SOAR

1. Click **Configuration** > **Credentials** > **Create Credential**.
2. Specify the following parameter values in the **Credential Editor**:
   a. **Internal Credential**

Parameter	Value
Type	Internal Credential
Name	Display name of credential set (For example, Forti Manager Credentials)
Username	User that was created for SOAR on Forti Manager
Password	Password of the user that was created for SOAR on Forti Manager
Private Key	Empty

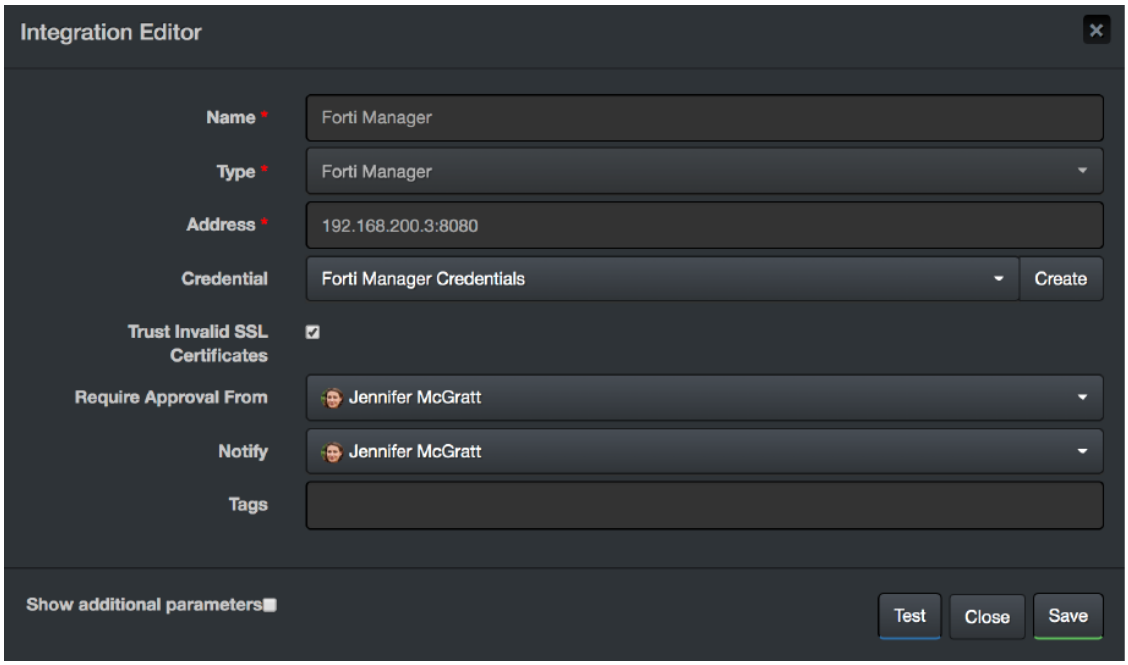
   b. **Credential Store:**

Parameter	Value
Type	External Credential
Name	Name of the credential with pull path of the safe on store

3. Click **Configuration** > **Integrations** > **Create Integration**.
4. Specify the following parameter values in the **Configuration** form:

Parameter	Value
Name	Display name of FortiMail integration on SOAR
Type	Forti Manager
Address	Address of the integration (in the following format: https://192.168.2.2:8080)

Credential	Name of the credential set created on step 2 (For example, Forti Manager Credentials)
Trust Invalid SSL Certificates	Select this if Forti Manager's certificate is self signed or is not recognized by browsers
Require Approval From	Select users from the list who can provide approval before executing actions on this integration
Notify	Select users from the list to notify when SOAR performs an action on this integration



5. Click **Test** to test the integration.
6. Click **Save** to complete the integration.

Additional Notes

Commands to be run on Forti Gate firewall devices are defined as Advanced Action Script. To access the default scripts navigate to **Configuration** > **Customization Library**.

Integration Guide for Fortinet FortiSandbox

Integration Overview

Fortinet Sandbox is a zero-day malware behavior analysis system. It enables organizations to defend against advanced threats such as ransomware by integrating various Fortinet technologies and other security products. Or is used as an extension to their on-premise security architectures to leverage complete control.

Integration Capabilities

ArcSight SOAR has the following integration capabilities with Fortinet Sandbox:

- Query File Hash
- Analyze File
- Analyze URL

Use Case: Investigating Suspicious Files

During the investigation of a suspicious endpoint behavior, SOAR integrated with Fortinet Sandbox analyzes the behavior of potential malware and hashes and URLs detected on suspicious network traffic. This investigation can either be performed automatically within a playbook or manually by an analyst.

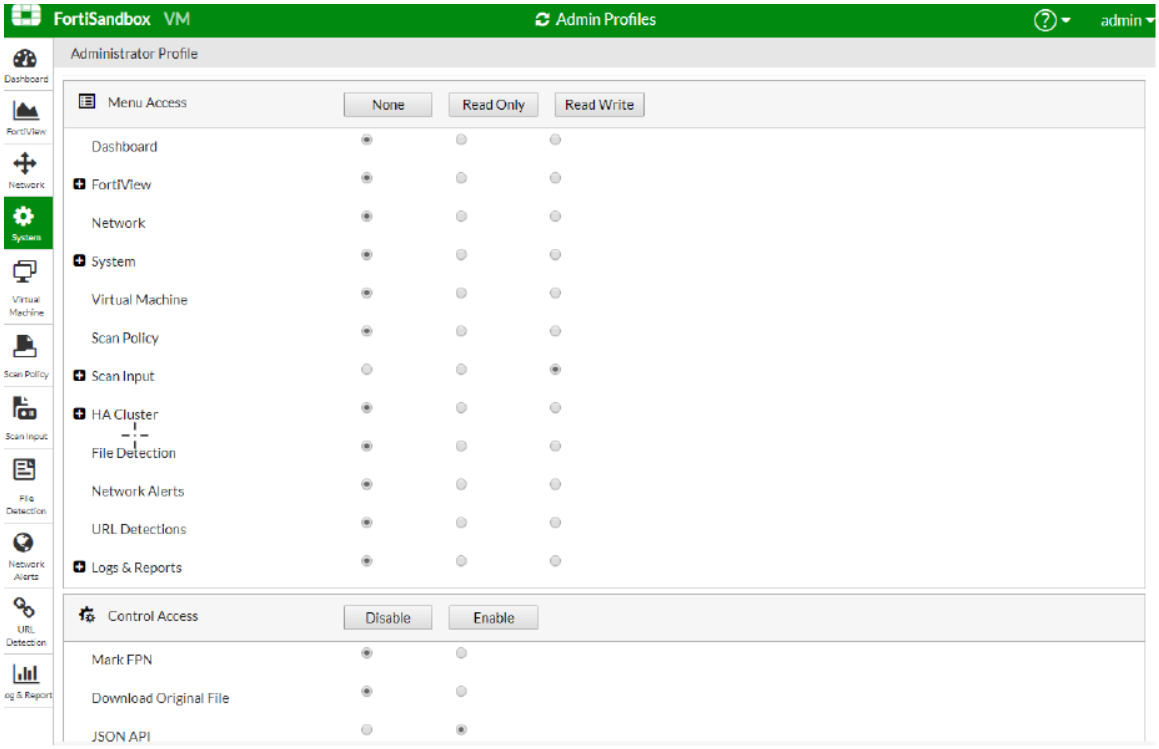
Configuration

Prerequisites

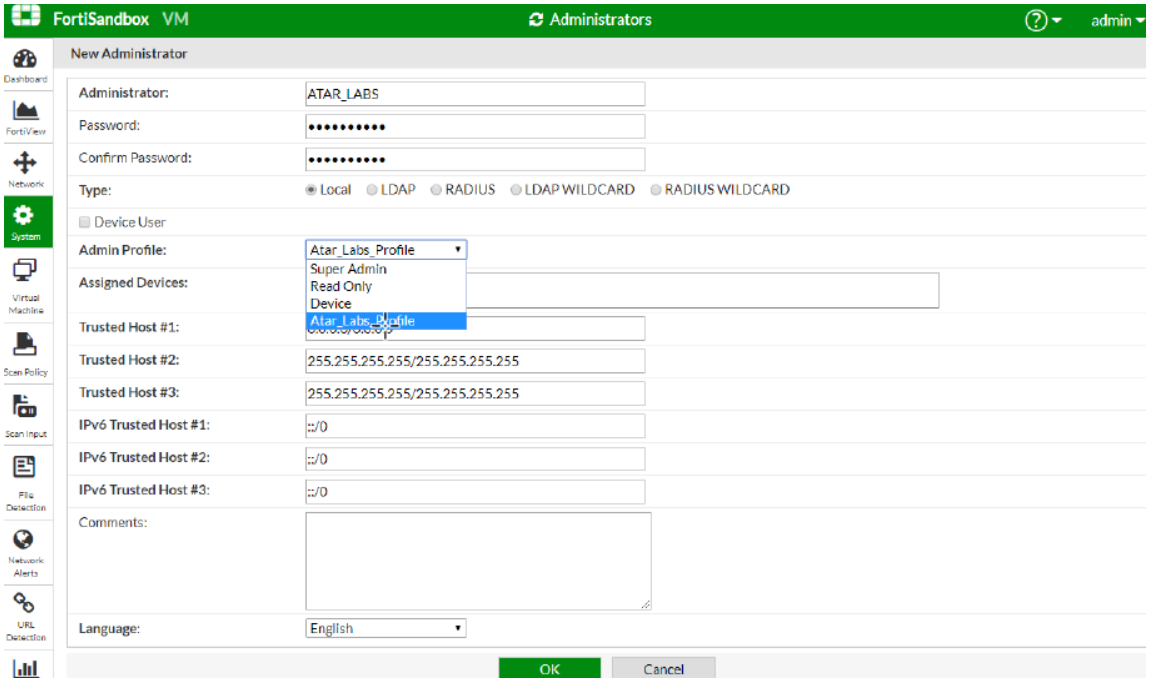
- Fortinet Sandbox 3.1.0 version as SOAR supports it
- Access to tcp port 443 as SOAR connects to Fortinet Sandbox API using HTTPS
- A user account is required for SOAR to connect to Fortinet Sandbox

Configuring Fortinet Sandbox

1. Navigate to **System** > **Admin Profiles**.
2. Create an Admin Profile with **Read/Write permission** for **SCAN INPUT** and select **Enable** for **JSON API**.



3. Navigate to **System** > **Administrators**.
4. Create an **Administrator** account with the profile that is created in the previous step and specify the following values:
   - **Administrator:** SOAR_LABS
   - **Password:** <Specify the password>
   - **Confirm Password:** <Confirm the password spcified in the Password field>
   - **Type:** Select **Local**
   - **Admin Profile:** <Specify the profile name>



Configuring SOAR

1. Click **Configuration** > **Credentials** > **Create Credential**.
2. Specify the following parameter values in the **Credential Editor**:
   a. **Internal Credential**

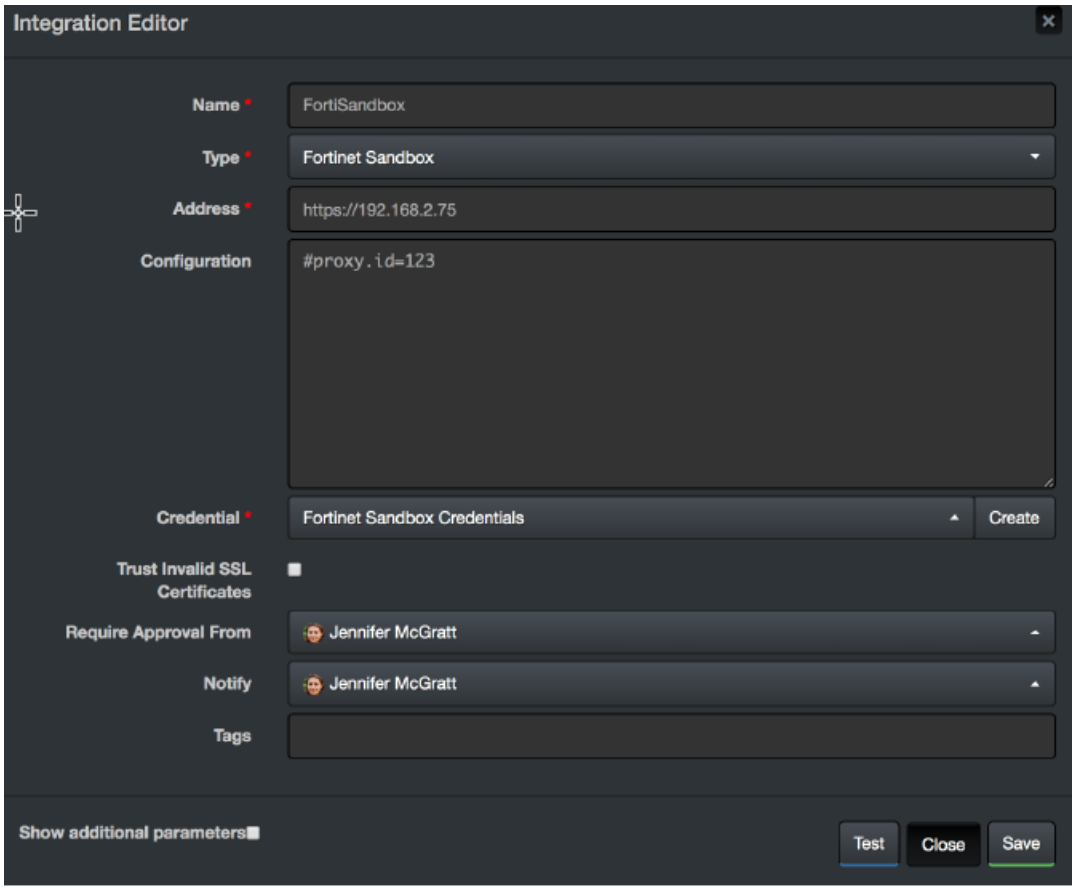
Parameter	Value
Type	Internal Credential
Name	Display name of credential set (For example, Fortinet Sandbox Credentials)
Username	User that was created on Fortinet Sandbox for SOAR
Password	Password of the user that was created for SOAR on Fortinet Sandbox
Private Key	Empty

   b. **Credential Store:**

Parameter	Value
Type	External Credential
Name	Name of the credential with pull path of the safe on store

3. Click **Configuration** > **Integrations** > **Create Integration**.
4. Specify the following parameter values in the **Configuration** form:

Parameter	Value
Name	Display name of Fortinet Sandbox integration on SOAR
Type	Fortinet Sandbox
Address	Address of the integration (in the following format: https://192.168.2.75)
Configuration	Specify the following configuration parameters: `#proxy.id=5442`
Credential	Name of the credential set created on step 2 (For example, Fortinet Sandbox Credentials)
Trust Invalid SSL Certificates	Select this if Web UI's certificate certificate is self-signed or is not recognized by browsers
Require Approval From	Select users from the list who can provide approval before executing actions on this integration
Notify	Not Applicable



5. Click **Test** to test the integration.
6. Click **Save** to complete the integration.

Additional Notes

Fortinet Sandbox supports the following compressed file types:

.tar, .z, .xz, .gz, .tar.gz, .tgz, .zip, .bz2, .tar.bz2, .tar.Z, .7z, .rar, .lzh, .ace

Integration Guide for FTP Server

Integration Overview

ArcSight SOAR uses FTP Servers to put or transfer files to remote machines using incident scope.

Integration Capabilities

Action

- Put File

Configuration

Prerequisites

- Access to File Transfer Protocol or SFTP as SOAR connects to FTP Server using it
- A user's credential

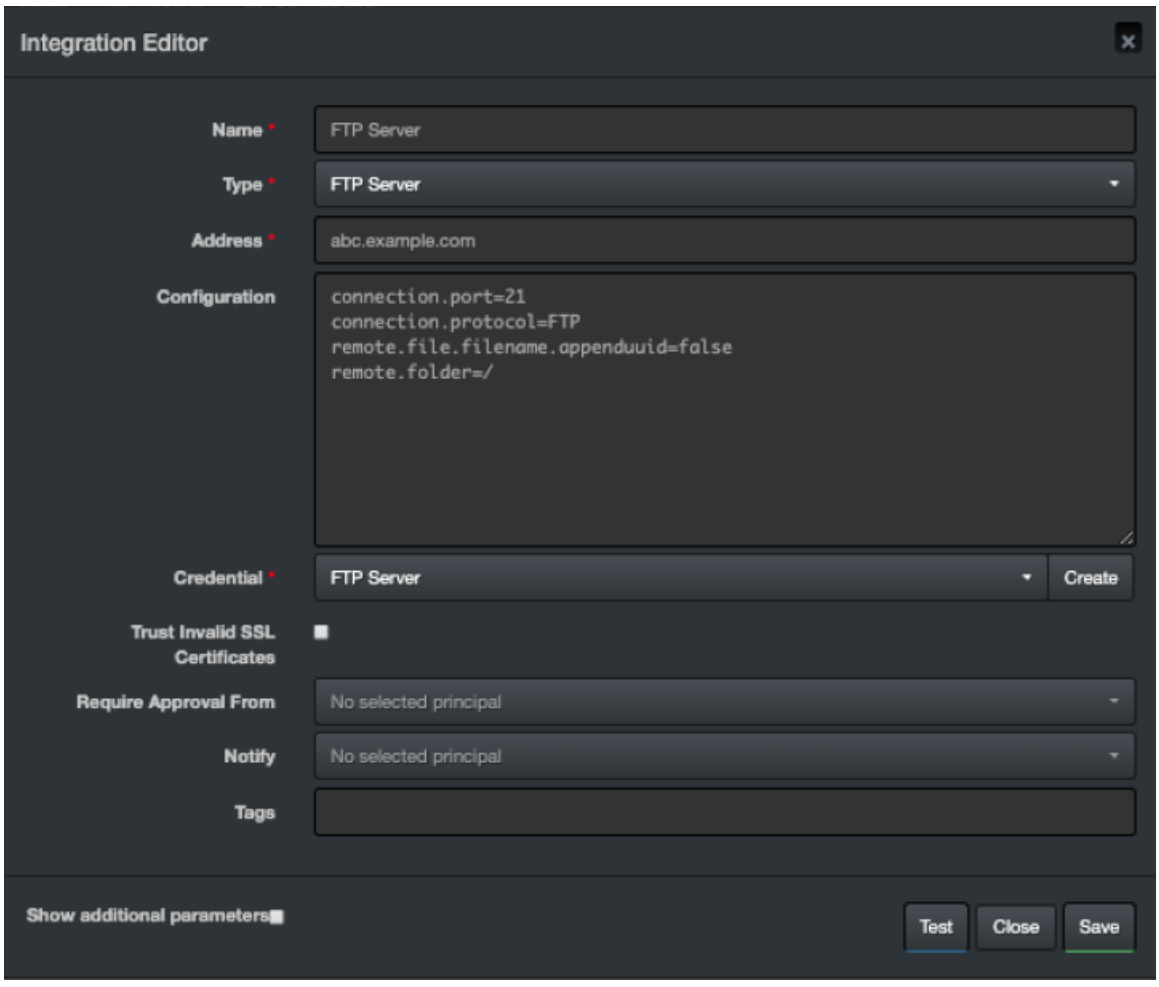
Configuring SOAR

1. To create the integration, navigate to **Configuration** > **Integrations**.
2. Specify the following parameter values in the **Integrations Editor** form.

Parameter	Value
Name	Display name of the integration
Type	FTP Server
Address	Address of the integration (in the format: 1.1.1.1 or abc.example.com)

Configuration	Specify the following configuration parameters:
	`connection.port is the listening port of the FTP/SFTP service running.`
	`connection.protocol could be FTP or SFTP.`
	`remote.file.filename.appenduuid specifies whether the UUID will be appended to the filename. It can be either “true” or “false”.`
	`remote.folder is the folder relative to the FTP home directory.`

Credential	Credential that was defined for this integration under the **Credentials** menu
Trust Invalid SSL Certificates	Select this if device’s certificate is self-signed or is not recognized by browsers
Require Approval From	Select users from the list who can provide approval before executing actions on this integration
Notify	Select users from the list to notify when SOAR performs an action on this integration



3. Click **Test** to test the integration.
4. Click **Save** to complete integration.

Integration Guide for Generic HTTP SMS Gateway

Integration Overview

ArcSight SOAR uses Generic HTTP SMS (Short Message Service) Gateway to send SMS.

Integration Capabilities

- None

Configuration

Configuring Generic HTTP SMS Gateway

- Access to **File HTTPS** service as SOAR uses it to connect to Generic HTTP SMS Gateway
- A SOAR user account

Configuring SOAR

1. To create the integration, navigate to **Configuration** > **Integrations**.
2. Specify the following parameter values in the **Integrations Editor** form.

Parameter	Value
Name	Display name of the integration
Type	Generic HTTP SMS Gateway
Address	Address of the integration (in the following format: 1.1.1.1 or abc.example.com)

Configuration	Specify the following configuration parameters:
	`http.method = POST`
	`http.auth.enabled = false`
	`params.jobID = ${credential.privateKey}`
	`params.url = http://dev.swh.soarlabs.io/atar/`
	`params.username = ${credential.username}`
	`params.text = ${text}`
	`params.gsmNumber = ${recipient}`
	`http.header.User-Agent = SOAR`
	`http.header.Content-Type = application/x-www-form-urlencode`
	`sms.stripCountryCode = +90`

Credential	Credential that was defined for this integration under the **Credentials** menu
Trust Invalid SSL Certificates	Select this if device's certificate is self-signed or is not recognized by browsers
Require Approval From	Select users from the list who can provide approval before executing actions on this integration
Notify	Select users from the list to notify when SOAR performs an action on this integration

3. Click **Test** to test the integration.
4. Click **Save** to complete integration.

Integration Guide for HTTP Proxy

Integration Overview

ArcSight SOAR uses HTTP proxies to access information on HTTP based integrations.

- SOAR's internal license management, automatic upgrade features use a proxy to request a new license and download new SOAR versions or additional content.
- To enable the features through a proxy, grant access to https://delivery.atarlabs.io.
- If you have selected an online proxy connection method to enable the features, see the **Registration Modes** section.
- Some alert sources can use a proxy to retrieve alerts and intelligence feeds. See the **respective integration guides** for configuring the proxy.
- Some integrations are capable of accessing resources on the Internet or other networks through a proxy device configuration. See the **respective integration guides** for configuring the proxy.

Configuration

Prerequisites

- Access to proxy service for SOAR
- A SOAR user account to connect to proxy if proxy authentication enabled

Configuring HTTP Proxy

HTTP Proxy software needs no further configuration.

Configuring SOAR

1. Click **Configuration** > **Credentials** > **Create Credential**.
2. Specify the following parameter values in the **Credential Editor**:

a. **Internal Credential**

Parameter	Value
Type	Internal Credential
Name	Display name of credential set (for example, HTTP Proxy Credentials)
Username	User that was created on HTTP proxy software for SOAR
Password	Password of the user that was created on HTTP proxy software for SOAR
Private Key	Empty

b. **Credential Store:**

Parameter	Value
Type	External Credential
Name	Name of the credential with pull path of the safe on store

3. Click **Configuration** > **Integrations** > **Create Integration**.
4. Specify the following parameter values in the **Configuration** form:

Parameter	Value
Name	Display name of HTTP Proxy integration on SOAR
Type	HTTP Proxy
Address	Address of the integration (in the following format: https://192.168.1.3:8081)

Configuration	Specify the following configuration parameters:
	`# Supported values: none, basic, ntlm`
	`# none: No authentication even if credential is provided`
	`# basic: basic HTTP digest authentication and`
	`# ntlm (NTLM authentication if your proxy validates the credential through`
	`# its Microsoft Active Directory integration.`
	`# For NTLM, username in credential should be specified like: username@domain`
	`authentication.type=basic`
	`# URL to use when testing availability of this proxy integration.`
	`# Defaults to the value of HttpProxyCheckURL configuration parameter.`
	`availabilitycheck.url=http://www.soarlabs.io`

Credential	Name of the credential set created on step 2 (For example, HTTP Proxy Credentials)
Trust Invalid SSL Certificates	Select this if Web UI's certificate certificate is self-signed or is not recognized by browsers
Require Approval From	Select users from the list who can provide approval before executing actions on this integration
Notify	Select users from the list to notify when SOAR performs an action on this integration

5. Click **Test** to test the integration.
6. Click **Save** to complete the integration.

Additional Notes

For SOAR to perform Automatic Update Checks, navigate to **Configuration** > **Parameters** and set `ProxyIntegrationIdForAutomaticUpdateCheck`.

Integration Guide for IBM Security QRadar

Integration Overview

IBM QRadar Security Information and Event Management (SIEM) helps security teams to detect and prioritize threats across the enterprise. It provides insights that enable teams to respond quickly and reduce the impact of incidents. QRadar consolidates log events, network flow data from devices, endpoints, and applications distributed throughout the network, and correlates, aggregates related events into a single alert to accelerate incident analysis and remediation.

Integration Capabilities

ArcSight SOAR has the following integration capabilities with IBM QRadar:

- Ingest Offenses and Events as Alert
- Retrieve Base Events
- Retrieve Flow Data
- Close related offenses
- Update related offenses
- Run Ariel Query

Use Case #1: Automated Alert Ingestion and Triage

SOCs need to use multiple products and platforms to detect, investigate, and mitigate the attacks. During the life cycle of an incident, analysts spend time performing repetitive tasks such as log enrichment, data analysis to prioritize the incident. However, the time spent on prioritizing might have resolved the issue itself. After ingesting offenses from IBM QRadar, SOAR automatically gets all the underlying associated logs to populate and enrich the incident context by getting valuable information from third party products and platforms. Based on enriched incident context, SOAR helps SOC prioritize and resolve incidents effectively.

Use Case #2: Interactive Investigation of Alerts

While playbooks automate the investigation and mitigation process of the incidents, an investigation requires interactive or manual activities. These activities enable gathering additional evidence by checking employees' views on the case and tasks performed on multiple security technologies to resolve the incident. This entire process is time-consuming for

analysts. SOAR provides actionable information and visibility to SOC analysts about the attack by enriching the incident. SOAR lets the analysts decide about the task based on the evidence gathered.

Configuration

Prerequisites

Following are the two main integration methods supported by SOAR:

- SOAR periodically connects to IBM QRadar's API to get new offenses and updates (default)
- IBM QRadar is configured to forward offenses and updates

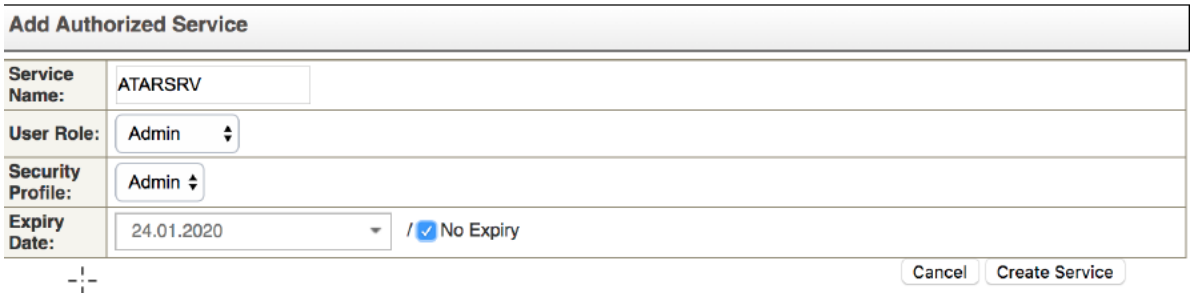
Note: The first method is recommended as it requires less configuration, is more robust and durable against network related problems.

- IBM QRadar version 7.2.7 Build 20160519230548 or later
- Access to tcp port 8090 for the network traffic from IBM QRadar to SOAR
- Access to tcp port 443 as SOAR connects to IBM QRadar API using HTTPS
- API token for SOAR to connect to IBM Radar

Configuring IBM QRadar

1. To add an authorized service, navigate to **Admin** > **User Management** > **Authorized Services**.

   Note: Save the **Authentication Token** generated for this service.

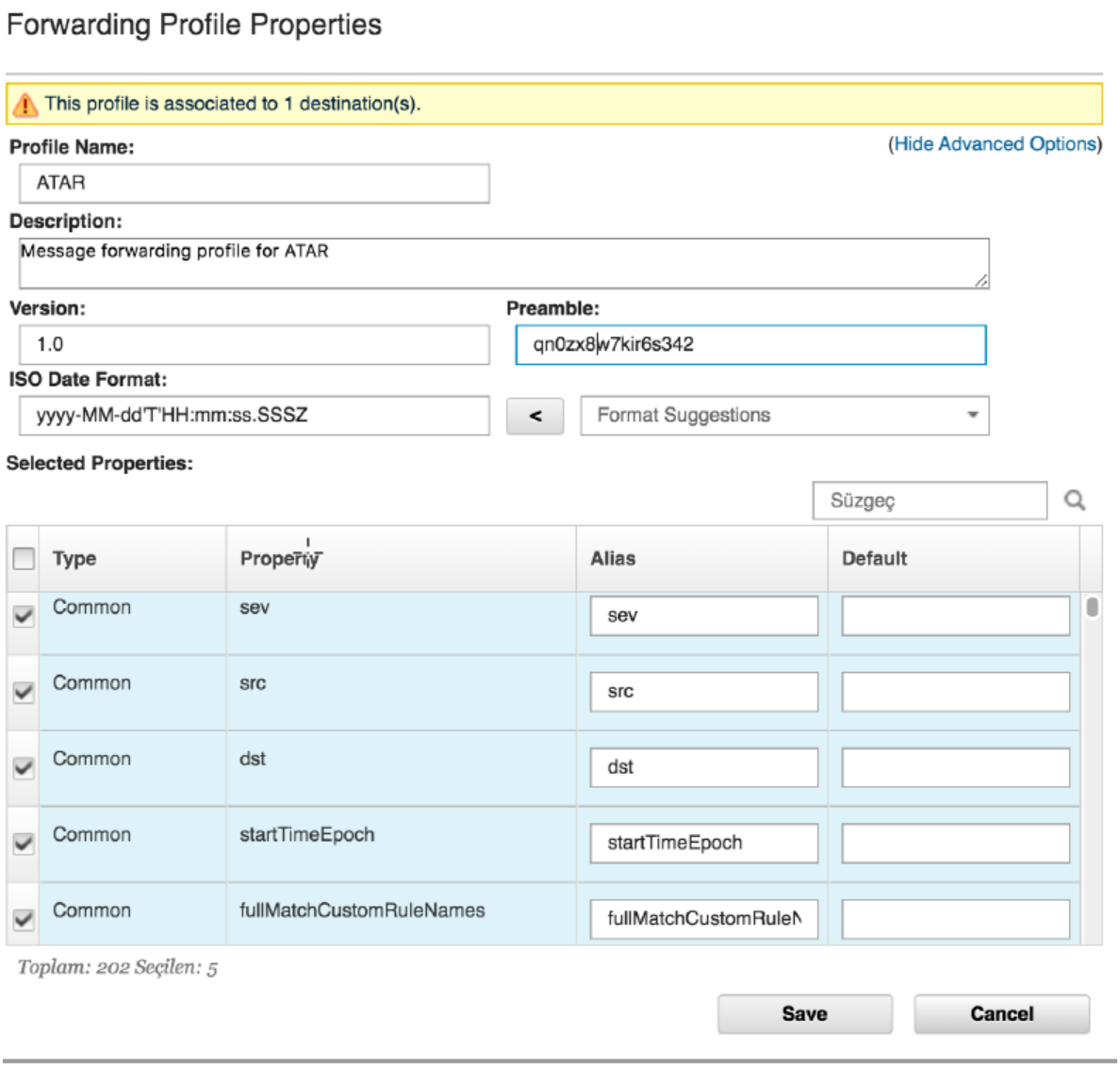
   

   Note: You can skip Steps 2 and 3 if you need SOAR to fetch offenses or events periodically using the SyncTask method. However, not required to forward those to SOAR.

2. To add a new forwarding destination, navigate to **Admin** > **System Configuration** > **Forwarding Destinations**.

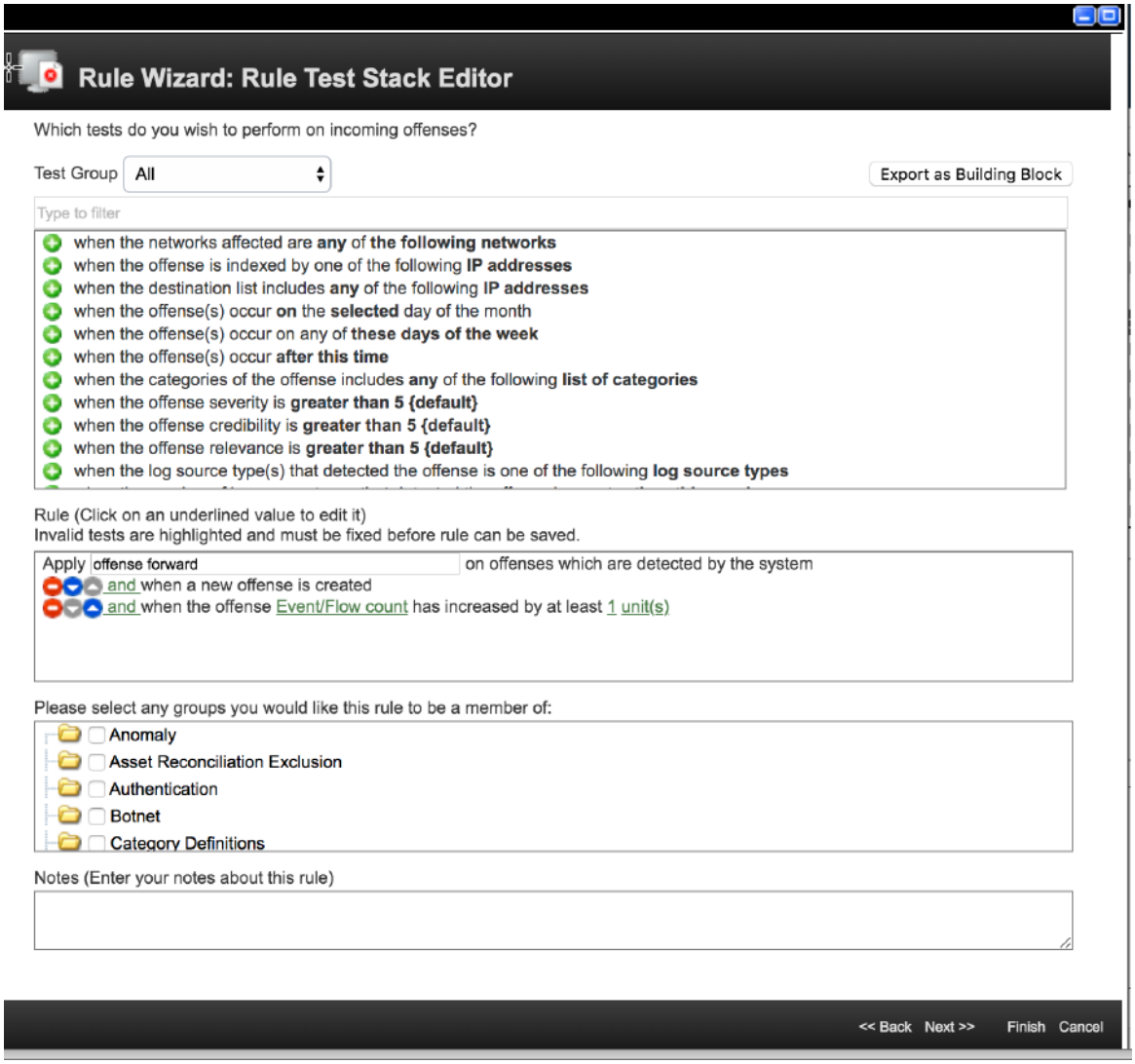
   Note: While configuring the forwarding destination, you must configure a forwarding profile.

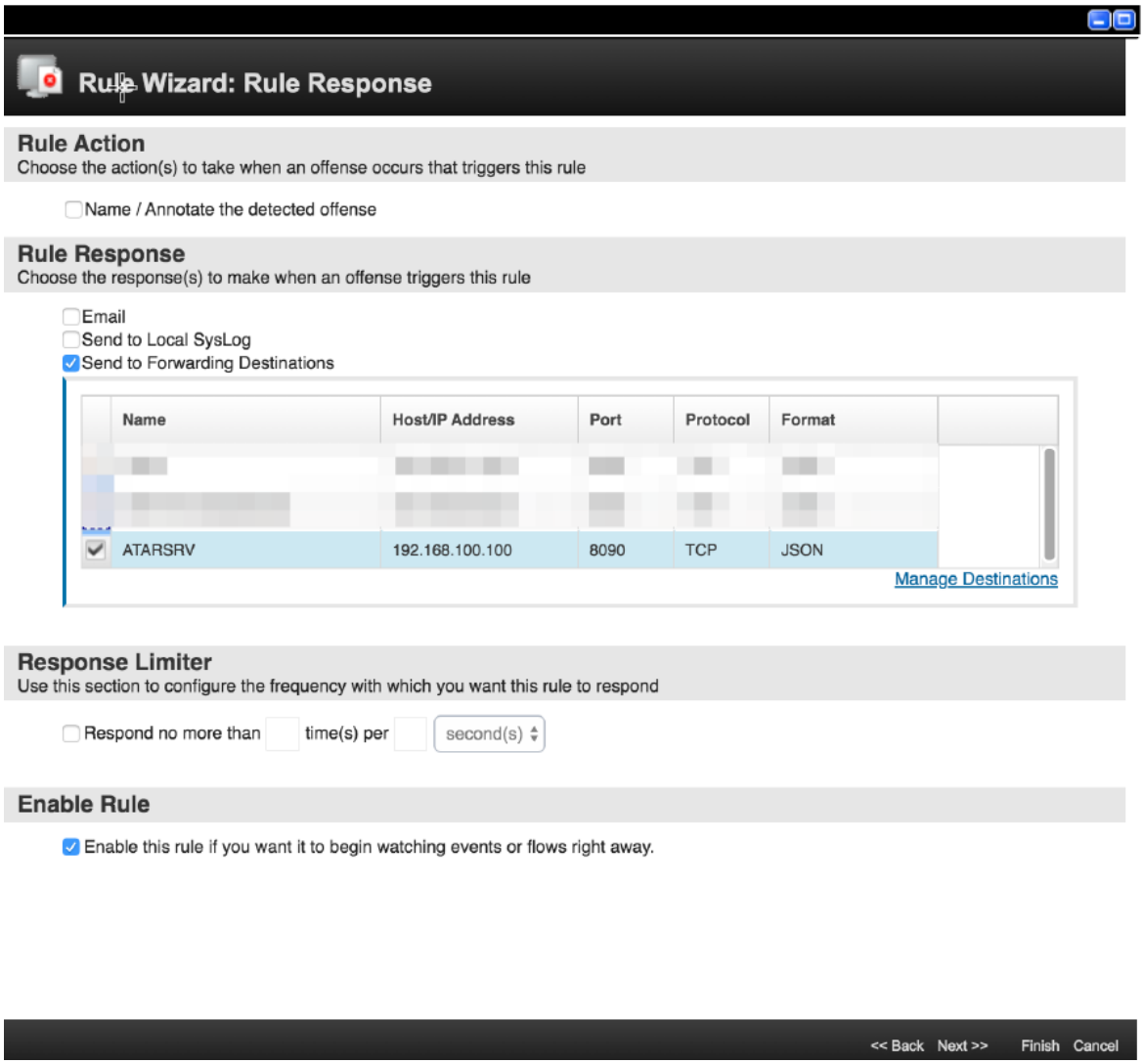
3. Specify the following values in the **Forwarding Profile Properties:**
   - **Preamble value:** < Specify a value for SOAR to use it to validate the forwarded messages>
   - **offense-based alerts:** select the **Payload** property.
   - **event-based alerts:** select the following JSON properties:
     src, dst, sev, startTimeEpoch, fullMatchCustomRuleNames



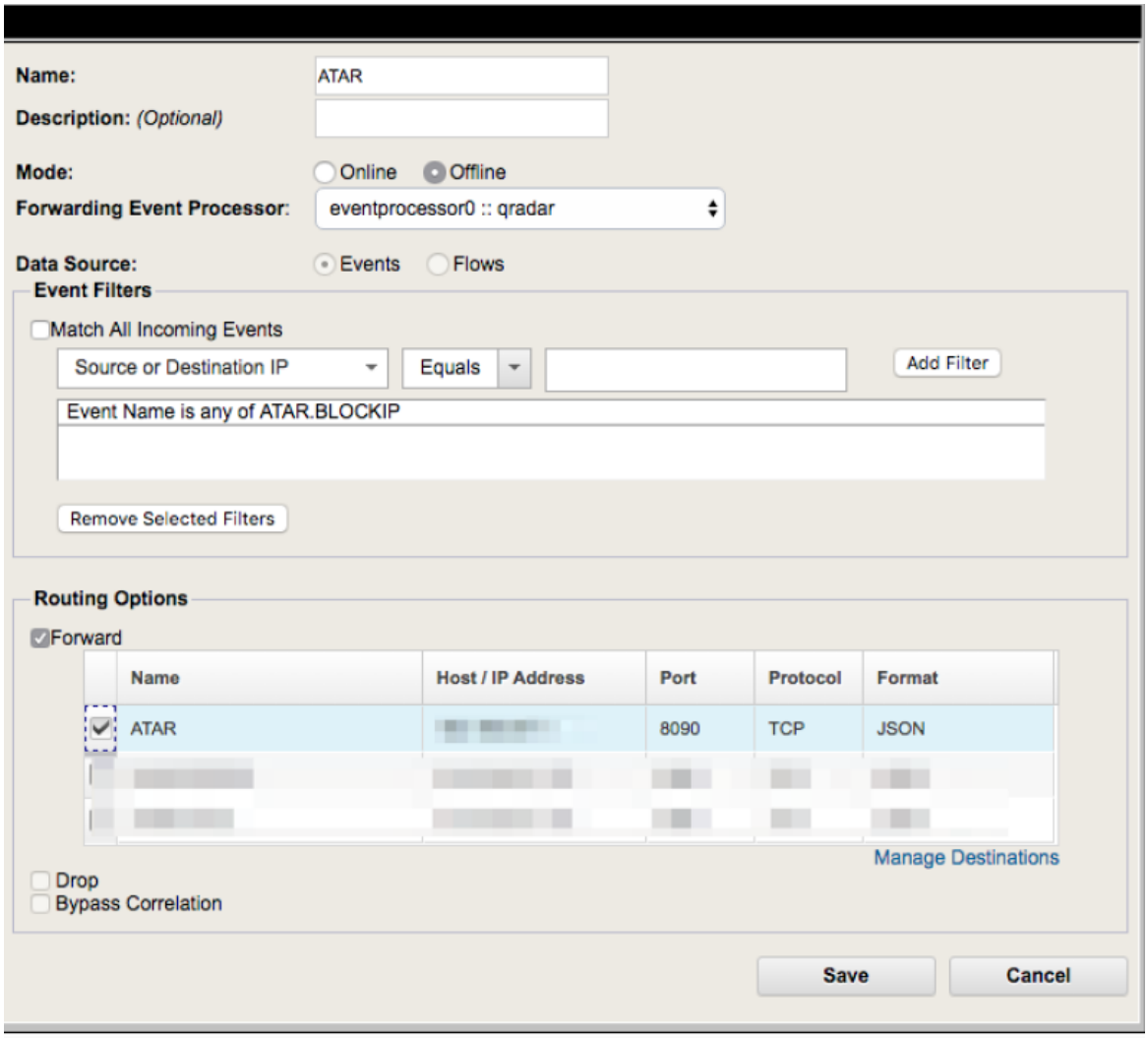
4. To add a new offense rule, navigate to **Offenses** > **Rules**.

> **Tip:** From a list of offenses and events, you can select individual events or offenses and edit the rules to be forwarded to SOAR.


Rule Wizard: Rule Test Stack Editor
Which tests do you wish to perform on incoming offenses?
Test Group All
Export as Building Block
Type to filter
when the networks affected are any of the following networks
when the offense is indexed by one of the following IP addresses
when the destination list includes any of the following IP addresses
when the offense(s) occur on the selected day of the month
when the offense(s) occur on any of these days of the week
when the offense(s) occur after this time
when the categories of the offense includes any of the following list of categories
when the offense severity is greater than 5 {default}
when the offense credibility is greater than 5 {default}
when the offense relevance is greater than 5 {default}
when the log source type(s) that detected the offense is one of the following log source types
Rule (Click on an underlined value to edit it)
Invalid tests are highlighted and must be fixed before rule can be saved.
Apply offense forward on offenses which are detected by the system
and when a new offense is created
and when the offense Event/Flow count has increased by at least 1 unit(s)
Please select any groups you would like this rule to be a member of:
Anomaly
Asset Reconciliation Exclusion
Authentication
Botnet
Category Definitions
Notes (Enter your notes about this rule)
<< Back Next >> Finish Cancel



5. (Optional) To forward events in bulk to SOAR, navigate to **Admin** > **Routing Rules** > **Add**.
6. Configure a listner to forward messages from QRadar to SOAR with the following values and define a routing rule:
   - **Mode:** select <**offline**> to ensure guaranteed delivery
   - Filter messages based on a rule name or the other criteria



Configuring SOAR

1. Click **Configuration** > **Credentials** > **Create Credential**.
2. Specify the following parameter values in the **Credential Editor**:

a. **Internal Credential**

Parameter	Value
Type	Internal Credential
Name	Display name of credential set (for example, IBM QRadar Credentials)
Username	Empty
Password	Empty
Private Key	API Token of Authorized web service that was created on IBM QRadar

b. **Credential Store:**

Parameter	Value
Type	External Credential
Name	Name of the credential with pull path of the safe on store

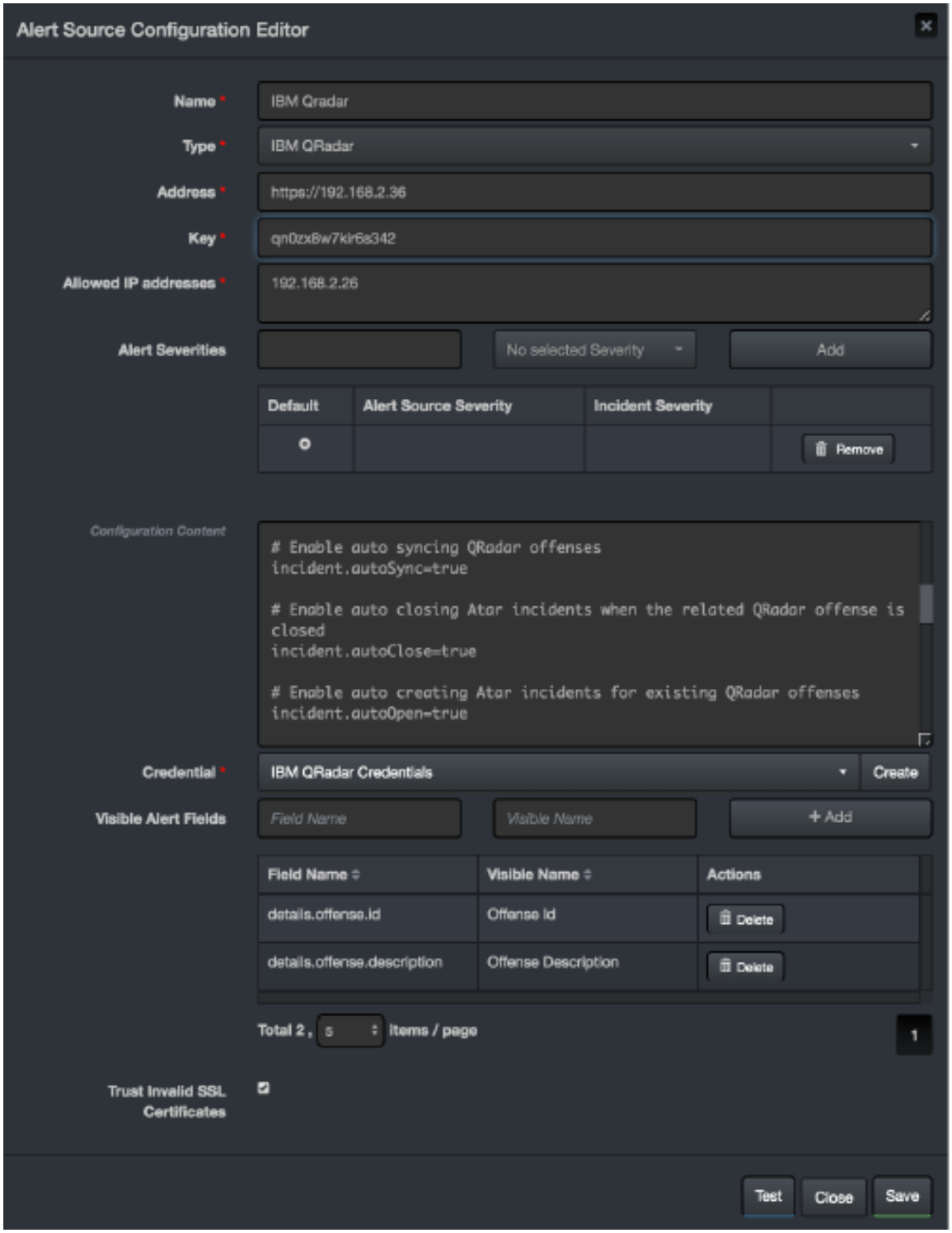
Configuring IBM QRadar as Alert Source

1. Click **Configuration** > **Alert Source** > **Create Alert Source Configuration**.
2. Specify the following parameter values in the **Configuration** form:

Parameter	Value
Name	Display name of IBM QRadar Alert Source on SOAR
Type	IBM QRadar
Address	Address of the IBM QRadar (in the following format: https://192.168.2.36)
Key	Preamble value set in **Forwarding Profile configuration** **Important:** SOAR QRadar listener discards any data without this key.
Allowed IP Addresses	IP address of the IBM QRadar instance
Alert Severities	Mapping of alert severity values to SOAR incident severities

Configuration	Specify the following configuration parameters:
	`# Direct DB access is required for IBM QRadar version 7.3.1p1 and older.`
	`For OffenseRule-based #directdbaccess.enabled=true`
	`#directdbaccess.jdbcurl=jdbc:postgresql://hostname:5432/qradar`
	`#directdbaccess.credential=`
	`# The offense field to be used as severity in SOAR (default: magnitude)`
	`#soar.severity.field=`
	`# Enable/disable base events activity in the incident timeline`
	`#enable.baseevent.activity=false`
	`# Additional event table columns, including custom event properties, seperated by commas`
	`#event.custom.columns=Catalog, Application Category`
	`# Enable auto syncing QRadar offenses`
	`#incident.autoSync=false`
	`# Enable auto closing Atar incidents when the related QRadar offense is closed`
	`#incident.autoClose=false`
	`# Enable auto creating SOAR incidents for existing QRadar offenses`
	`#incident.autoOpen=false`
	`# Maximum time to wait for search results from QRadar. If unset, value of`
	`# QRadarEnrichmentDefaultTimeout parameter will be used.`
	`#timeout=60`
	`# Maximum polling wait delay when checking the result of searches on QRadar.`
	`# If unset, value of QRadarEnrichmentMaxPollDelay will be used.`
	`#maxpolldelay=1000`
	`# Scope fields to be extracted from base events and/or correlated events (field1:CATEGORY:ROLE,`

	`# CATEGORY is any of: EMAIL_ADDRESS, HASH, HOST, MAC_ADDRESS, NETWORK_ADDRESS, # COMPUTER_NAME, UNKNOWN, URL, USERNAME, PROCESS # ROLE is any of: OFFENDER, IMPACT, RELATED# # Note: The fields in the baseevent.scope example below are always extracted by default. # Note: Extraction with same field name overrides the default one. # Note: Extraction with different field name does not override the default behaviour and extracted # Note: Field names must start with / character # # Example: baseevent.scope=/sourceip:NETWORK_ADDRESS:OFFENDER, /sourcemac:MAC_ADDRESS:OFFENDER, # baseevent.scope= # # Example: correlated.scope=/sourcev6:NETWORK_ADDRESS:OFFENDER # correlated.scope= # configure how far (in minutes) into the past this enrichment will look. #cache.reusing.duration=20 # How far (in days) into the past SOAR will look for offenses at the initial sync task # If not provided, SOAR will use 14 days by default #days.to.look.back.at.initial.sync=14`
Credential	Name of the credential set created on step 2 (For example, IBM QRadar Credentials)
Trust Invalid SSL Certificates	Select this if Web UI's certificate certificate is self-signed is or is not recognized by browsers
Visible Alert Fields	Define the alarm fields to be displayed on Incident Management Service Desk



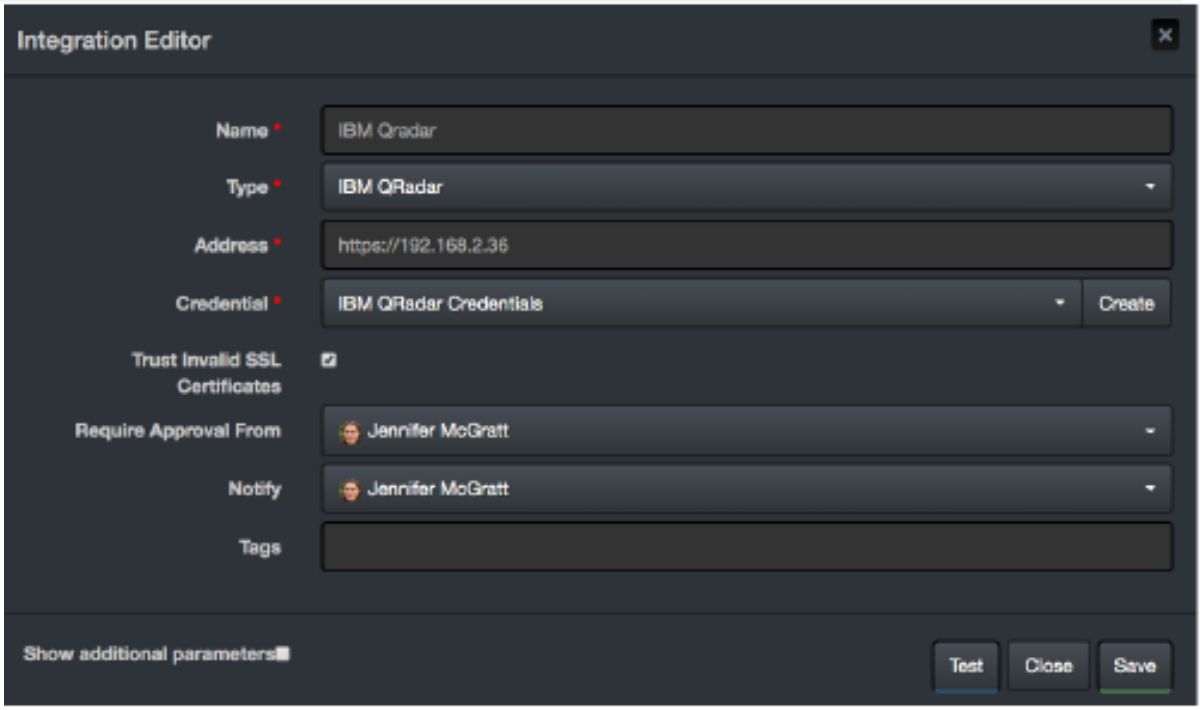
3. Click **Test** to test the integration.
4. Click **Save** to complete the integration.
5. If you have not configured `incident.autoSync=true` and want to forward offenses to SOAR from QRadar, navigate to **Configuration** > **Parameters**.

6. Select the value of **QRadarListenerEnabled** to true.

Configuring IBM QRadar as Integration

1. Click **Configuration** > **Integrations** > **Create Integration**.
2. Specify the following parameter values in the **Configuration Form**:

Parameter	Value
Name	Display name of IBM QRadar integration on SOAR
Type	IBM QRadar
Address	Address of the integration (in the following format: https://192.168.2.36)
Credential	Name of the credential set created on step 2 (For example, IBM Qradar Credentials)
Trust Invalid SSL Certificates	Select this if server's certificate is self signed or is not recognized by browsers
Require Approval From	Select users from the list who can provide approval before executing actions on this integration
Notify	Select users from the list to notify when SOAR performs an action on this integration



3. Click **Test** to test the integration.
4. Click **Save** to complete the integration.

Additional Notes

- You can use the following configuration parameters to finetune the integration:

Parameter Name	Description	Default Value
LEEFLogVersion	Log Event Extended Format (LEEF) version for QRadar	1.0
QRadarAPIRESTDeleteQueueConcurrency	Consumer count for QRadar Ariel API search deletion (Maximum thread count that ATAR utilize in order to delete search job).	1
QRadarAPIRESTDeleteSearchReadTimeout	QRadar Ariel API's search deletion read timeout in minutes (In case of unresponsive search job deletion session, ATAR closes the connection after this many minutes).	15
QRadarAPIRetryHTTPStatusCodes	QRadar API retry HTTP status codes comma separated	500,502,503
QRadarAPIRetryMaxSeconds	QRadar API max retry time in seconds. 0 for no retry	0
QRadarAPIRetrySleepSeconds	QRadar API retry sleep in seconds	5
QRadarAutoEnrichDayLimit	Auto enrich tries for getting base events on last n days (ATAR will not try to get base events older than that many days).	120
QRadarAutoEnrichEnabled	Enable QRadar auto-enrichment with base-event data	False
QRadarDefaultClosingReasonId	Reason id for closing offense on QRadar	1
QRadarEnrichmentDefaultReadTimeout	QRadar enrichment read timeout in seconds (In case of unresponsive search job creation session, ATAR closes the connection after this many second).	900
QRadarEnrichmentDefaultTimeout	QRadar enrichment timeout in seconds	60
QRadarEnrichmentMaxPollDelay	QRadar enrichment maximum poll delay in milliseconds.	1000
QRadarListenerAutoEnrichLimit	Maximum number of base events to retrieve per QRadar alert	10000
QRadarListenerAutoEnrichOverlap	Base event retrieval time interval overlap in milliseconds	0
QRadarListenerEnabled	Enable QRadar listener	False
QRadarListenerInboundRequestTimeout	QRadar listener inbound request timeout in millisecond	5000
QRadarListenerMaxMessageSize	Maximum message size limit for messages received from QRadar (byte)	16384
QRadarListenerMaxRetrySeconds	QRadar listener queue max message retry in seconds	1800
QRadarListenerPort	QRadar listener port	8090
QRadarListenerProcessingDelay	Delay in milliseconds before processing QRadar alerts	0
QRadarListenerQueueConcurrency	Count of consumer threads for listening QRadar offenses (Maximum thread count that ATAR utilize in order to process QRadar offenses. This should be smaller than connection limit defined in device configuration).	5
QRadarListenerSingleIncidentPerOffense	Create single incident per offense (If true, create single alert/ticket per offense. If false, create an alert/ticket for each rule related to an offense)	True
QRadarListenerThreadPoolCoreSize	QRadar listener thread pool core pool size (0 = unlimited)	0
QRadarListenerThreadPoolKeepAlive	QRadar listener thread pool keep-alive seconds (ignored if core pool size = 0)	60
QRadarListenerThreadPoolMaxSize	QRadar listener thread pool maximum size (ignored if core pool size = 0)	20
QRadarListenerThreadPoolQueueCapacity	QRadar listener thread pool queue capacity (ignored if core pool size = 0)	1000
QRadarMissingOffenseMaxTryCount	Maximum number of try to fetch a missing offense (Maximum retry count for unresponded offenses. This parameter is only for SyncTask usage).	12
QRadarMissingOffensePageSize	Page size of Missing Offense Query (Batch count of unresponded offenses that is being retried. This parameter is only for SyncTask usage).	20
QRadarMissingOffenseRange	Maximum acceptable range of missing offense ids (Maximum offense count that defines unresponsive QRadar API. This parameter is only for SyncTask usage).	100
QRadarMissingOffenseTryAgainInMinutes	Minutes to wait for retrying a QRadar missing offense (Retry interval for unresponded offenses. This parameter is only for SyncTask usage).	60
QRadarSyncLookBehindMinutes	Minutes to look behind to offense in QRadar SyncTask (This parameter defines search time window for each offense sync. This parameter is only for SyncTask usage).	30
QRadarSyncPeriod	Period in seconds to sync QRadar offenses	60
AlertSourceAPIBaseEventRetryCount	Alert source base event enrichment retry count after a failed base event enrichment (Maximum count of unresponded base event fetch retry).	3
AlertSourceAPIBaseEventRetryInMinutes	Minutes to wait for alert source base event enrichment retry after a failed base event enrichment (First retry interval for unresponded base event fetching process. Will be multiplied by 5 between every try).	10

- Following are the two main methods are supported by SOAR:
  a. In the first method, SOAR periodically connects to IBM QRadar REST-API through Authorized Service API Token to get new offenses and updates using the SyncTask

method. This integration requires no SOAR listener configuration.

b. The second method, configuring IBM QRadar to forward offenses and updates to SOAR listener, highly relies on an error-less network operation. If any error occurs on the network connection, IBM QRadar doesn't retry to transmit the unsent offense and offense updates, which means loss of alert and alert updates.

- SOAR handles the following IBM Security QRadar alerts:
  a. **Offense-based alerts:** Creates a single alert for a single offense that h alert can have one or more offender and impact addresses.
  b. **OffenseRule-based alerts:** Creates one or more alerts for an offense. A separate alert gets created for each rule that triggered the offense and can have one or more offender and impact addresses.
  c. **Event-based alerts:** Creates an alert for a single event that can have one or more offender and impact addresses.
- In SOAR, the configuration parameter QRadarListenerSingleTicketPerOffense (default = true), controls the choice between Offense-based and OffenseRule-based alerts.
- Event-based alerts are created independently from Offense-based or OffenseRule-based alerts. If an Event Rule or Routing Rule is used to forward events from IBM Security QRadar to SOAR, SOAR recognizes that the message describes an event and creates a matching alert in SOAR.
- If you have used an Offense Rule to forward offenses from IBM Security QRadar to SOAR, SOAR recognizes the message as an offense and creates or updates alerts accordingly.
- Offenses are potentially long-lasting. When new information gets added, the existing Offenses and OffenseRule-based alerts in SOAR gets updated in IBM Security QRadar. For example, when additional offender or impact addresses get added to an alert, SOAR executes the actions, updates this new information. However, for Event-based alerts, updating information doesn't take place.
- To collect Offense and OffenseRule-based alerts information, SOAR needs access to the IBM Security QRadar REST-API. Due to API restrictions for IBM QRadar 7.3.1 p1 and older versions, SOAR requires access to the IBM Security QRadar database. Hence, create a SOAR user with permissions to select information on OffenseRule-based alerts from table OFFENSE_CRE_AGG.
- SOAR allows users to define a Closing Reason while they close an offense on IBM QRadar. You can create a new Closing Reason or select one from the list of reasons existing on IBM QRadar. If none selected, `QRadarDefaultClosingReasonId` sets the default value.
- The IBM QRadar integrates with Alert Source that enables QRadar actions in the SOAR Administration Interface to retrieve the list of existing Closing Reasons. If multiple instances are there, the alert source gets selected to ensure that SOAR connects and modifies the correct QRadar alert..

- All the base event extraction rules must start as shown in configuration (for example /destinationip:NETWORK_ADDRESS:IMPACT).
- A relation must be defined to not miss the BaseEvents between QRadar connection limit and concurrency in `QRadarListener.processMessage` method.
  a. **ATAR Version 2.16.1 or before:**

```
10 (concurrency max) +
1 (test) +
1 (buffer)
QRadar Configuration: ==> connection.limit=12
```

  b. **ATAR Version 2.16.2 or after:**

```
5 (QRadarListenerQueueConcurrency concurrency max) +
1 (delete queue QRadarApiRestDeleteQueueConcurrency) +
1 (test) +
1 (buffer)
QRadar Configuration: ==> connection.limit=8
```

- To fetch the base-events successfully, configure the same Timezones for Qradar and SOAR.

Integration Guide for IBM Security X-Force

Integration Overview

IBM X-Force Exchange is a cloud-based threat intelligence platform that enables users to research security threats, search attack indicators, aggregate actionable intelligence, and collaborate with peers.

Integration Capabilities

ArcSight SOAR has the following integration capabilities with IBM X-Force Exchange:

- DNS Records
- IP Report
- Malware for File Hash
- Send File for Analysis
- URL Report

Use Case: Investigating Phishing Campaigns

SOAR follows the user's email inbox for phishing reports and automatically creates an incident record on its service desk. While investigating the attack, SOAR extracts the sender address, IP address, URLs in the message body, files in the attachment, and checks with IBM X-Force Exchange if these attacks are previously analyzed. This investigation can either be performed automatically within a playbook or manually by an analyst.

Configuration

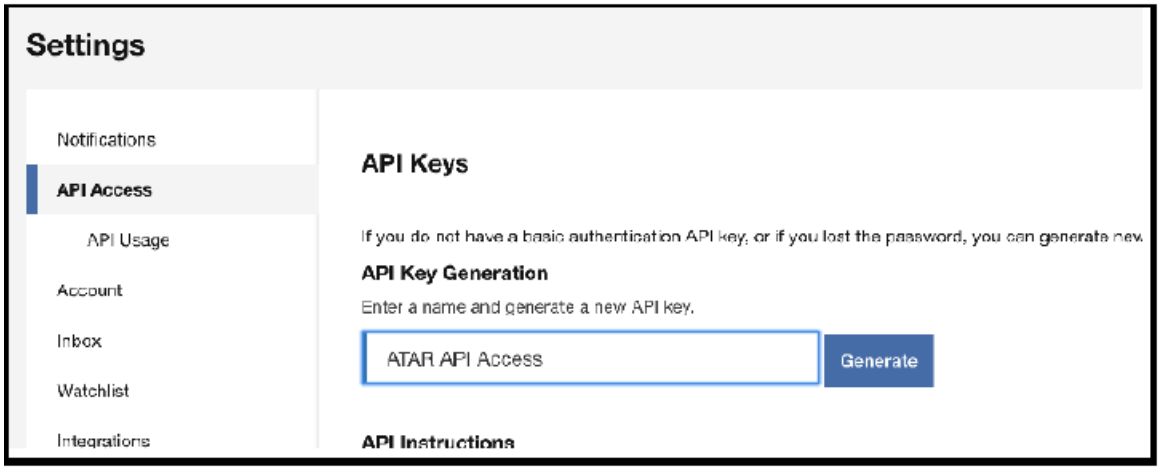
Prerequisites

- Access to https://api.xforce.ibmcloud.com (443/tcp port) for SOAR to connect to IBM X-Force Exchange API
- An API key for SOAR to connect to IBM X-Force Exchange

Configuring IBM X-Force Exchange

1. Log in to https://exchange.xforce.ibmcloud.com.
2. To create a new API key, navigate to **Settings** > **API Access**.

Note: Save the generated API key and the password.



Configuring SOAR

1. Click **Configuration** > **Credentials** > **Create Credential**.
2. Specify the following parameter values in the **Credential Editor**:
   a. **Internal Credential**

Parameter	Value
Type	Internal Credential
Name	Display name of credential set (For example, IBM X-Force Exchange Credentials)
Username	API Key created on IBM X-Force Exchange
Password	API Password for the key created on IBM X-Force Exchange
Private Key	Empty

   b. **Credential Store:**

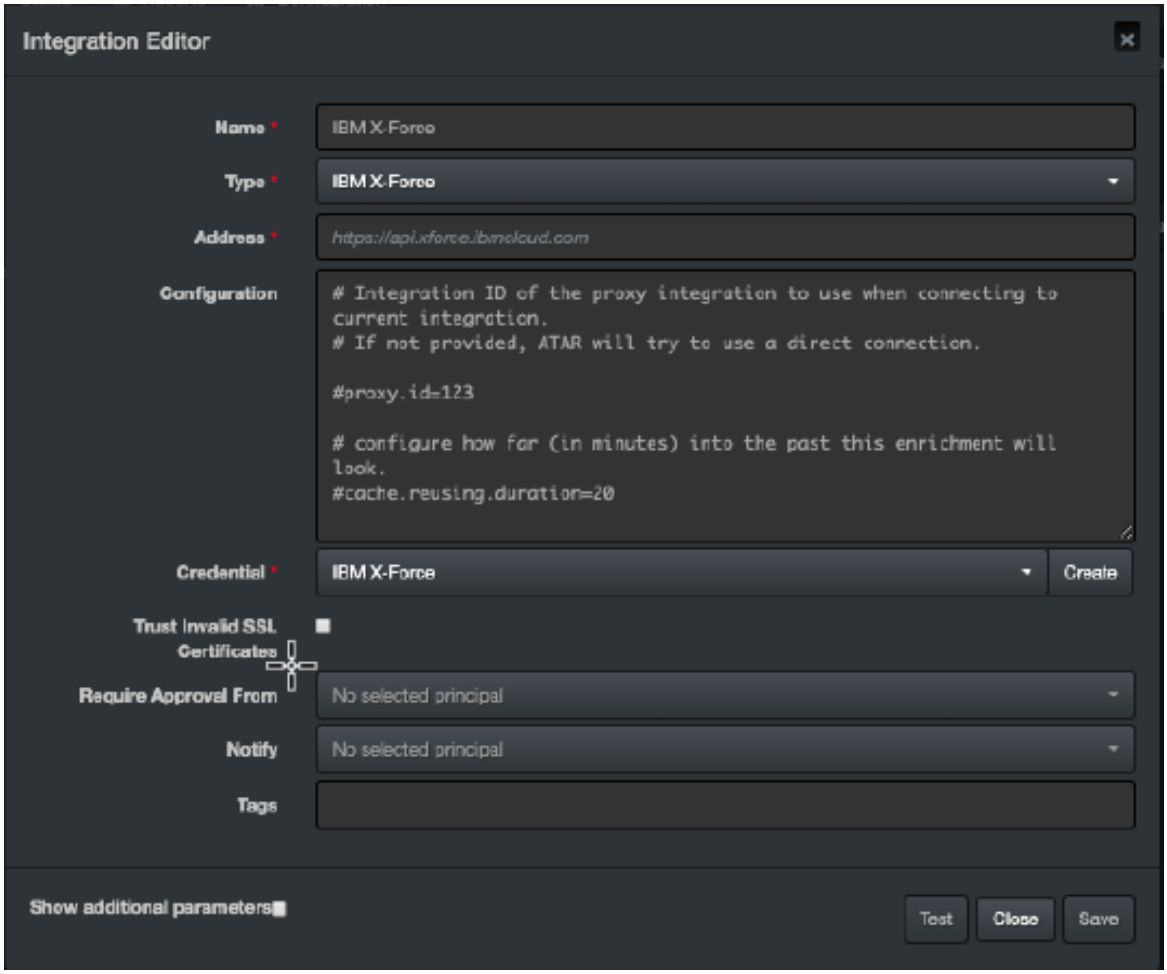
Parameter	Value
Type	External Credential
Name	Name of the credential with pull path of the safe on store

3. Click **Configuration** > **Integrations** > **Create Integration**.
4. Specify the following parameter values in the **Configuration** form:

Parameter	Value
Name	Display name of IBM X-Force Exchange integration on SOAR
Type	IBM X-Force Exchange
Address	Address of the integration (https://api.xforce.ibmcloud.com)

Configuration	Specify the following configuration parameters:
	`# Integration ID of the proxy integration`
	`to use when connecting`
	`# to current integration.`
	`# If not provided, SOAR will try to use`
	`a direct connection.`
	`#proxy.id=123`
	`# configure how far (in minutes) into the`
	`past this enrichment will look.`
	`cache.reusing.duration=60`

Credential	Name of the credential set created on step 2 (For example, IBM XForce Exchange Credentials)
Trust Invalid SSL Certificates	Select this if Web UI's certificate is self-signed or is not recognized by browsers
Require Approval From	Select users from the list who can provide approval before executing actions on this integration
Notify	Select users from the list to notify when SOAR performs an action on this integration



5. Click **Test** to test the integration.
6. Click **Save** to complete the integration.

Integration Guide for Infoblox DNS Firewall

Integration Overview

Infoblox DNS Firewall defends DNS servers from the comprehensive range of DNS-based attacks while maintaining service availability and business continuity. The Grid Manager web interface provides access to the appliance for network and IP address management.

Integration Capabilities

ArcSight SOAR has the following integration capabilities with Infoblox DNS Firewall:

- Block IP address (No Data)
- Block IP Address (No Such Domain)
- Block Host (No Data)
- Block Host (No Such Domain)
- Substitute DNS A Record

Use Case: Blocking malicious IP addresses on DNS

SOAR integrates with Infoblox DNS firewall to block malicious IP addresses and hosts on DNS firewall to stop malware attacks and protect users. These actions can either be performed automatically within a playbook or manually by an analyst.

Configuration

Prerequisites

- Infoblox NIOS 8.4 version
- Access to tcp port 443 as SOAR connects to Infoblox DNS Firewall API
- A SOAR user account to connect Infoblox DNS Firewall

Configuring Infoblox DNS Firewall

1. Navigate to **Administration** > **Administrators** > **Admins**.
2. To add an account, specify the following values in the **Add Administrator Wizard**:

   Authentication Type: Local

Login: <Specify the username>

Password: <Specify the password>

Confirm Password: <confirm the password specified in **Password** field>

Admin Group: Select *admin-group*

3. To create a new Response Policy Zone, navigate to **Data Management** > **DNS** > **Response Policy Zones**.

Configuring SOAR

1. Click **Configuration** > **Credentials** > **Create Credential**.
2. Specify the following parameter values in the **Credential Editor**:
   a. **Internal Credential**

Parameter	Value
Type	Internal Credential
Name	Display name of credential set (For example, Infoblox DNS FW Credentials)
Username	User created for SOAR on Infoblox DNS FW
Password	API Password for the key created for SOAR on Infoblox DNS FW
Private Key	Empty

   b. **Credential Store:**

Parameter	Value
Type	External Credential
Name	Name of the credential with pull path of the safe on store

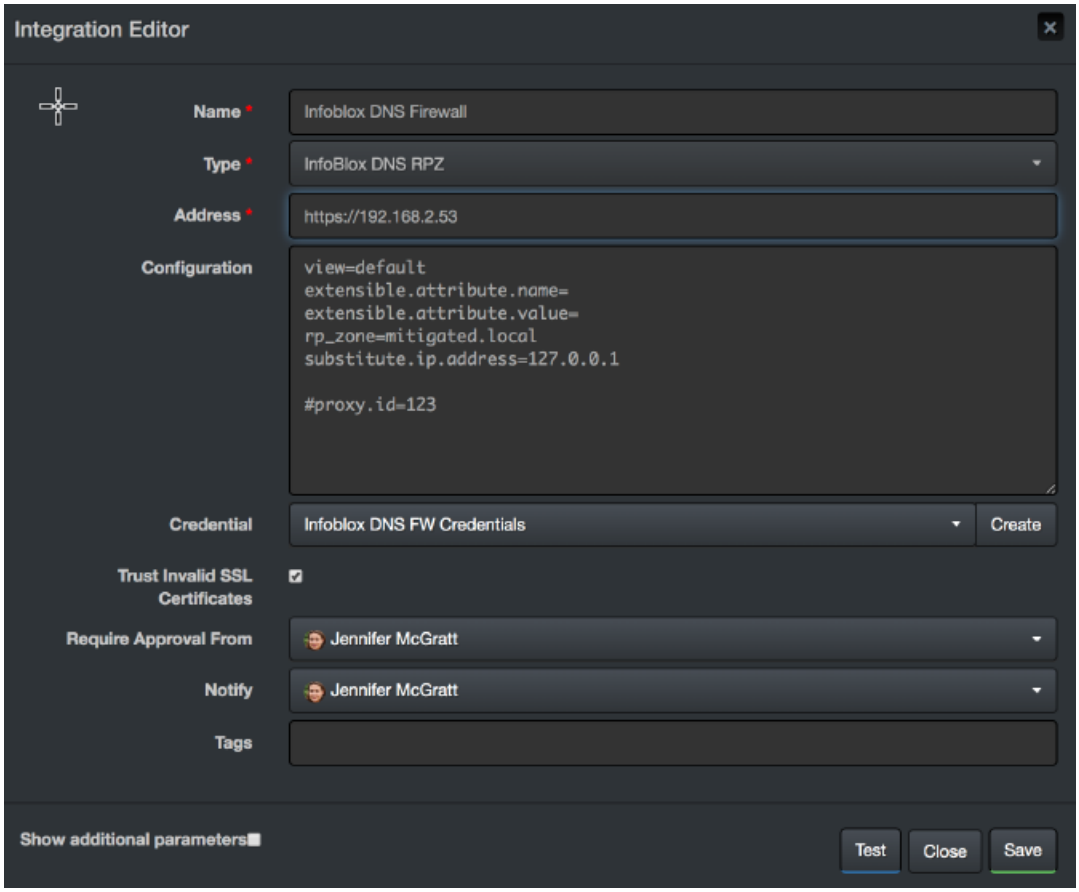
3. Click **Configuration** > **Integrations** > **Create Integration**.

4. Specify the following parameter values in the **Configuration** form:

Parameter	Value
Name	Display name of Infoblox DNS Firewall integration on SOAR
Type	Infoblox DNS Firewall
Address	Address of the integration (in the following format: https://192.168.2.53)

Configuration	Specify the following configuration parameters:
	`# Name of View under which rp_zone is located.`
	`view=default`
	`# Name of Response Policy Zone that SOAR will`
	`write block rules`
	`rp_zone=mitigated.local`
	`# Default name and value of extensible attrbute`
	`which SOAR uses to write comment for block`
	`extensible.attribute.name=`
	`extensible.attribute.value=`
	`# IP address that SOAR uses to substitute in`
	`DNS A records.`
	`substitute.ip.address=127.0.0.1`
	`#proxy.id=5442`

Credential	Name of the credential set created on step 2 (For example, Infoblox DNS FW Credentials)
Trust Invalid SSL Certificates	Select this if Web UI's certificate is self-signed or is not recognized by browsers
Require Approval From	Select users from the list who can provide approval before executing actions on this integration
Notify	Select users from the list to notify when SOAR performs an action on this integration.



5. Click **Test** to test the integration.
6. Click **Save** to complete the integration.

Additional Notes

Infoblox DNS Firewall allows blocking IP and host with only one rule type (either No Data or No Such Domain). If you try to block an IP or host that already got blocked with another rule type, you might get an error.

Integration Guide for Invictus USTA ThreatIntelligence

Integration Overview

Invictus USTA is a threat intelligence service which delivers cyber-threat insights in real time.

Integration Capabilities

- Ingest Threat Intelligence Feed as Alert
- Check Identity Leak
- Check Stolen Client Account
- Check Domain Info
- Check Hash Info
- Check IP Info
- Check URL Info
- Submit Bad Sender
- Submit Referer URL

Use Case: Blocking malicious URLs and IPs before they harm

ArcSight SOAR integrates with USTA intelligence feed to block malicious entities on your perimeter protection before they harm.

Use Case #2: Investigating Fraud and ID Theft

SOAR integrates with USTA Threat Intelligence to investigate fraud cases, possible ID theft, and cases of client account compromises.

Configuration

Prerequisites

- Access to https://usta01.invictuseurope.com/api/ (443/tcp port) for SOAR to connect to USTA API
- An API Key for SOAR to connect to Invictus USTA API

Configuring Invictus USPA

Invictus USTA requires no specific configuration.

Configuring SOAR

1. Click **Configuration** > **Credentials** > **Create Credential**.
2. Specify the following parameter values in the **Credential Editor**:
   a. **Internal Credential**

Parameter	Value
Type	Internal Credential
Name	Display name of credential set (For example,Invictus USTA Credentials)
Username	Empty
Password	Empty
Private Key	API Key obtained from Invictus USTA platform

   b. **Credential Store:**

Parameter	Value
Type	External Credential
Name	Name of the credential with pull path of the safe on store

Configuring Invictus USTA as Alert Source

1. Click **Configuration** > **Alert Source** > **Create Alert Source Configuration**.
2. Specify the following parameter values in the **Configuration** form:

Parameter	Value
Name	Display name of Invictus USTA Alert Source on SOAR
Type	USTA
Address	Address of the Invictus USTA Threat Intelligence Service (https://usta01.invictuseurope.com/api/)
Alert Severities	Mapping of alert severity values to SOAR incident severities

Configuration	Specify the following configuration parameters:
	`# Ignore events older than specified date. If empty, date based filtering is disabled.`
	`# Example: filterOlderThanDate=2017-01-01`
	`filterOlderThanDate=2020-01-10`
	`# Integration ID of the proxy integration to use when connecting to current source.`
	`# If not provided, SOAR will try to use a direct connection.`
	`#proxy.id=5523`

Credential	Name of the credential set just created. (For example, Invictus USTA Credentials)
Trust Invalid SSL Certificates	Select this if Web UI's certificate is self-signed or is not recognized by browsers
Visible Alert Fields	Define the alarm fields to be displayed on Incident Management Service Desk

3. Click **Test** to test the integration.
4. Click **Save** to complete the integration.

Configuring Invictus USTA as Integration

1. Click **Configuration** > **Integrations** > **Create Integration**.
2. Specify the following parameter values in the **Configuration** form:

Parameter	Value
Name	Display name of Invictus USTA integration on SOAR
Type	USTA
Address	Address of the Invictus USTA Threat Intelligence Service (https://usta01.invictuseurope.com)

Configuration	Specify the following configuration parameters:
	`# Integration ID of the proxy integration to use when connecting to current source.`
	`# If not provided, SOAR will try to use a direct connection.`
	`#proxy.id=5523`

Credential	Name of the credential set created on step 2 (For example, Invictus USTA Credentials)
Trust Invalid SSL Certificates	Select this if Web UI's certificate is self-signed or is not recognized by browsers
Require Approval From	Select users from the list who can provide approval before executing actions on this integration
Notify	Select users from the list to notify when SOAR performs an action on this integration.

3. Click **Test** to test the integration.
4. Click **Save** to complete the integration.

Additional Notes

USTA permits connection requests from specific network addresses for each customer. Hence, make sure to check the access permission by USTA before integration.

Integration Guide for JDBC(Database) Server

Integration Capabilities

ArcSight SOAR has the following integration capability with database servers:

- JDBC Query

Use Case: Querying HR Database

With this integration, while investigating an incident SOAR can run a query on HR database to see if they are logged on the user on a suspicious endpoint. This can either be performed automatically within a playbook or manually by an analyst.

Configuration

Prerequisites

- A database listener or service for SOAR to access.
- Create a DB user account for SOAR to run the SQL queries.

Configuring Database Server

Please contact database administrator for user account and access permissions.

Configuring SOAR

1. Click **Configuration** > **Credentials** > **Create Credential**.
2. Specify the following parameter values in the **Credential Editor**:

a. Internal Credential

Parameter	Value
Type	Internal Credential
Name	Display name of credential set (For example, JDBC Credentials)
Username	User account that was configured on database server
Password	Password for user account that was configured on database server
Private Key	Empty

b. Credential Store

Parameter	Value
Type	External Credential
Name	Name of the credential with pull path of the safe on store.

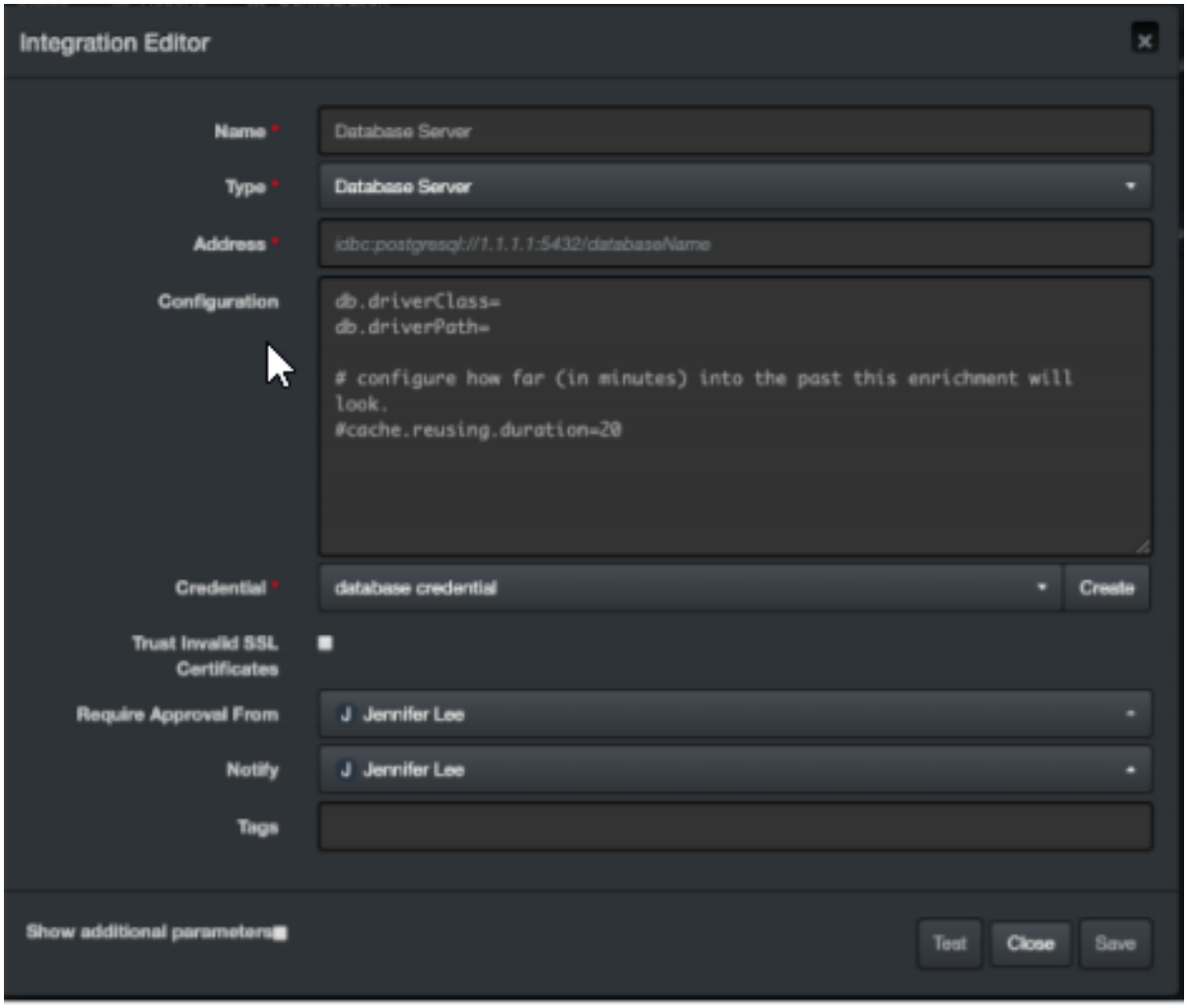
3. Click **Configuration** > **Integrations** > **Create Integration**.
4. Specify the following parameter values in the **Configuration** form:

Parameter	Value
Name	Display name of Database Server integration on SOAR
Type	Database Server
Address	Address of the integration (in the format jdbc:driverName://192.168.3.10:5432/databaseName).

Configuration	Specify the following configuration parameters:
	`# For MySQL: db.driverClass=com.mysql.jdbc.Driver`
	`# For Oracle: db.driverClass=oracle.jdbc.OracleDriver`
	`# For PostgreSQL: db.driverClass=org.postgresql.Driver`
	`# For MSSQL Server: db.driverClass=com.microsoft.sqlserver.jdbc.SQLServerDriver`
	`db.driverClass= db.driverClass=org.postgresql.Driver`
	`# Absolute path where you put the JDBC driver's JAR file.`
	`db.driverPath=`
	`# configure how far (in minutes) into the past this enrichment will look.`
	`cache.reusing.duration=30`

Credential	Name of the credential set created on step 2. (For example, Database Server Credentials).

Trust Invalid SSL Certificates	Select this if device's certificate is self-signed or is not recognized by browsers
Require Approval from	Select users from the list who can provide approval before executing actions on this integration
Notify	Select users from the list to notify when SOAR performs an action on this integration.



5. Click **Test**. The following pop up will be displayed if your credential and address are valid.
6. Click **Save** to complete integration.

Integration Guide for Juniper SRX Firewall

Integration Overview

SOAR uses Juniper SRX Firewall to block IP addresses on the network perimeter using the incident scopes.

Integration Capabilities

Action

- Block
- Custom Script

Configuration

Configuring Juniper SRX Firewall

- Access to SSH as SOAR connects to Juniper SRX Firewall integration using SSH
- A SOAR user with admin role

Configuring SOAR

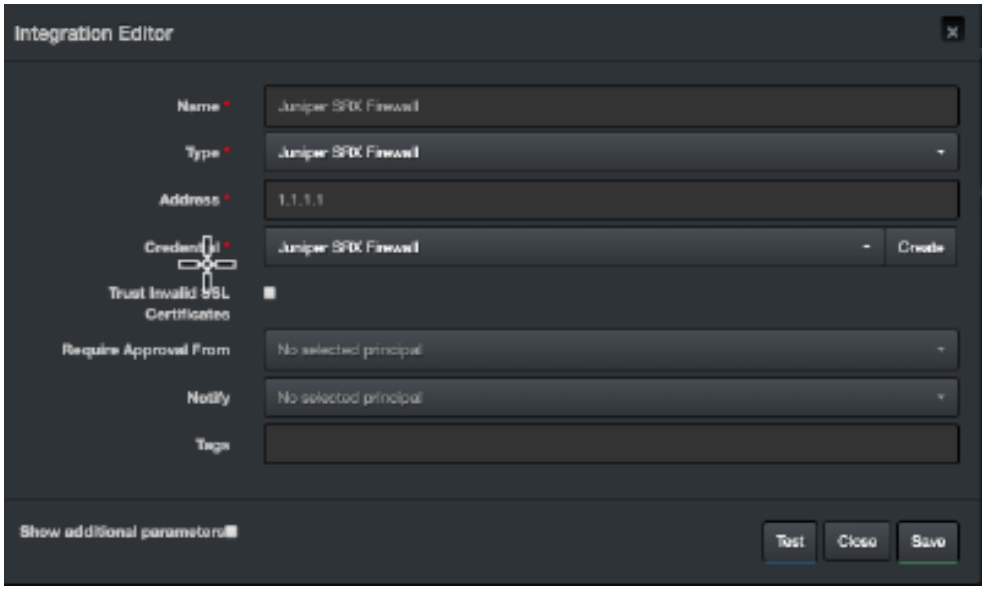
1. Click **Configuration** > **Integrations** > **Create Integration**.
2. Specify the following parameter values in the **Configuration** form:

Parameter	Value
Name	Display name of the integration
Type	Juniper SRX Firewall
Address	Address of the integration (in the following format: 1.1.1.1 or abc.example.com)
Credential	Name of the credential set created on step 2 (For example, FortiMail Credentials)

Trust Invalid SSL Certificates	Select this if Integrations's certificate is self-signed or is not recognized by browsers.
Require Approval From	Select users from the list who can provide approval before executing actions on this integration
Notify	Select users from the list to notify when SOAR performs an action on this integration

Note: You might have to review the integration actions defined and executed through the Juniper SRX Firewall related custom scripts in SOAR.

3. To find the following custom scripts, navigate to **Configuration** > **Custom Scripts**.
   - Juniper SRX Firewall Availability Check Default Template
   - Juniper SRX Firewall SSH Device Action (Block) Default Template



4. Click **Test** to test the integration.
5. Click **Save** to complete the integration.

Integration Guides for Kannel SMS Gateway

Integration Overview

Kannel is an open source SMS Gateway which is used widely for sending in either single or bulk SMS(Short Message Service). Kannel links HTTP based services to various SMS centers using various protocols.

Integration Capabilities

Supported Action Capabilities

Kannel SMS Gateway allows user notifications using SMS messages which was set when creating the Playbook involving this integration.

Configuration

Configuring Kannel SMS Gateway

- Configure the integration to send SMS messages.

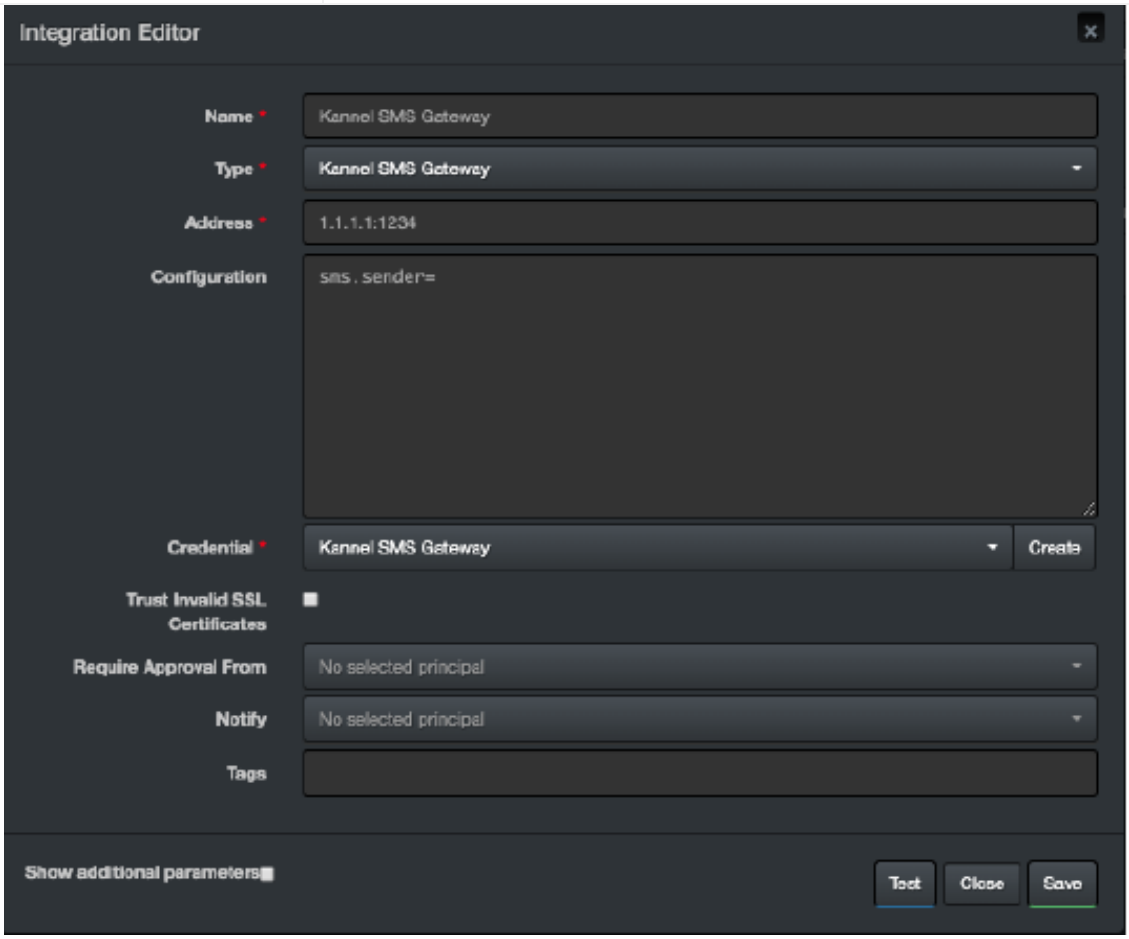
Configuring SOAR

Following are the steps to create the integration:

1. Navigate to **Configuration** > **Parameters**.
2. Configure **SMS Device** to be used as the ID of Kannel SMS Gateway integration.
3. To configure the integration, navigate to **Configuration** > **Integrations**.
4. Specify the following parameter values in the **Integration Editor**:

Parameter	Value
Name	Display name of Kannel SMS Gateway integration on SOAR
Type	Kannel SMS Gateway
Address	Address of the integration (in the following format: 1.1.1.1:1234)

Configuration	`sms.sender=`<Specify the value configured in the SMS Device field>
Credential	Name of the credential set created on step 2
Trust Invalid SSL Certificates	Select this if Integrations's certificate is self-signed or is not recognized by browsers.
Require Approval From	Select users from the list who can provide approval before executing actions on this integration
Notify	Select users from the list to notify when SOAR performs an action on this integration



5. Click **Test** to test the integration.
6. Click **Save** to complete the integration.

Integration Guide for Karmasis Infraskope

Integration Overview

Karmasis Infraskope is a SIEM platform which lets organizations to collect and correlate logs from systems and applications to create real-time alerts.

Integration Capabilities

ArcSight SOAR has the following integration capability with Karmasis Infraskope:

- Ingest Alerts

Use Case #1: Automated Alert Ingestion and Triage

Enterprise SOCs get hundreds of security alerts every day, and teams evaluate and prioritize those alerts. SOAR integrates with Karmasis Infraskope to help with the prioritization, investigation, and the remediation of incidents. A new incident gets created on SOAR Incident Management Service Desk, when an alert comes. SOAR allows analysts to investigate the case to take remedial actions.

Configuration

Prerequisites

- Karmasis Infraskope version *7.5.19.47
- Access to tcp port 443 as SOAR connects to Karmasis Infraskope API using HTTPS
- SOAR user account to connect to Karmasis Infraskope

Configuring Karmasis Infraskope

Dedicated username and password to access Karmasis Infraskope

Configuring SOAR

1. Click **Configuration** > **Credentials** > **Create Credential**.
2. Specify the following parameter values in the **Credential Editor** form:

a. **Internal Credential**

Parameter	Value
Type	Internal Credential
Name	Display name of credential set (For example, Karmasis Infraskope Credentials)
Username	Username you have created on Karmasis Infraskope
Password	Empty
Private Key	API Token you've obtained from Karmasis Infraskope

b. **Credential Store:**

Parameter	Value
Type	External Credential
Name	Name of the credential with pull path of the safe on store

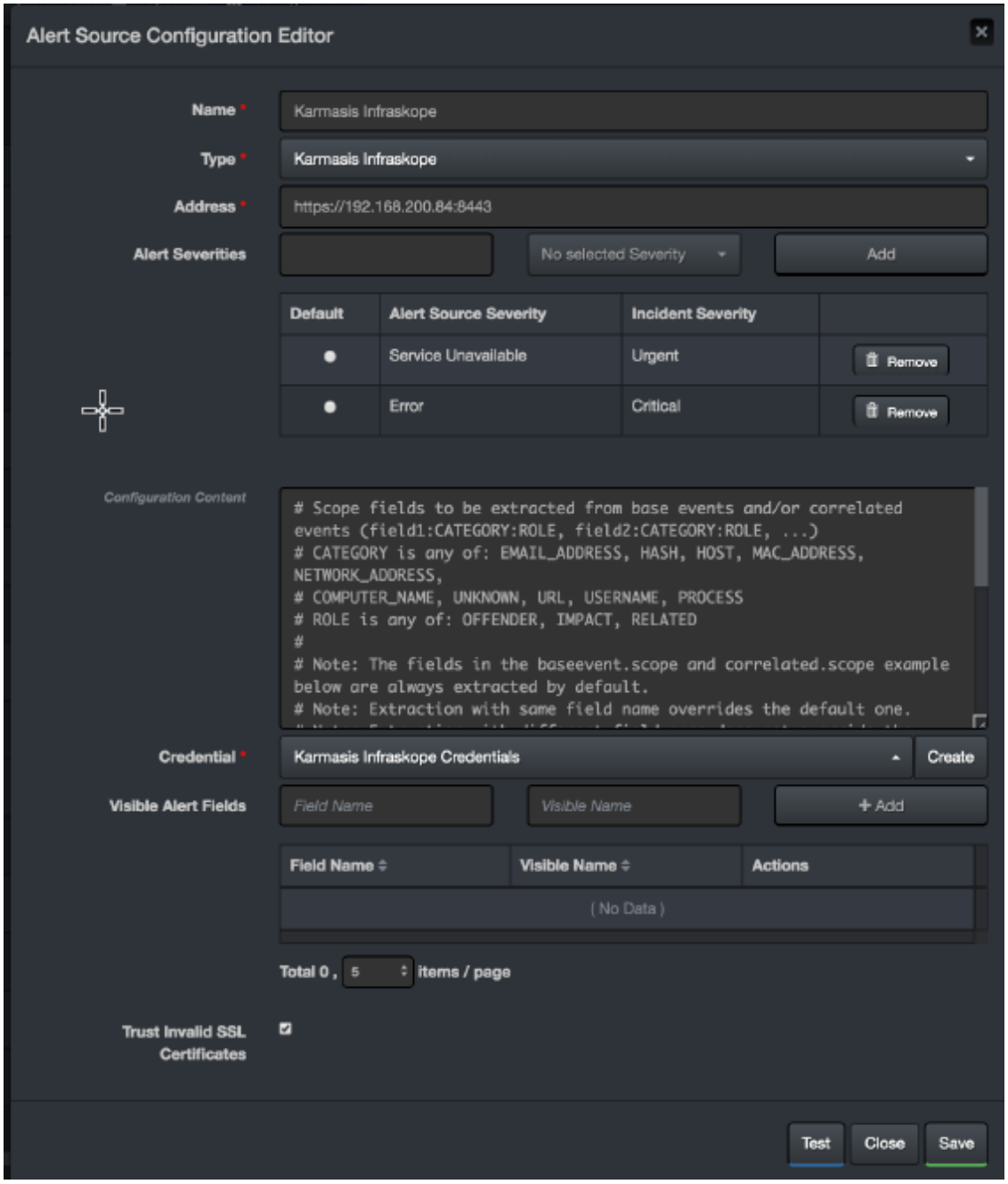
Configuring Karmasis Infraskope as Alert Source

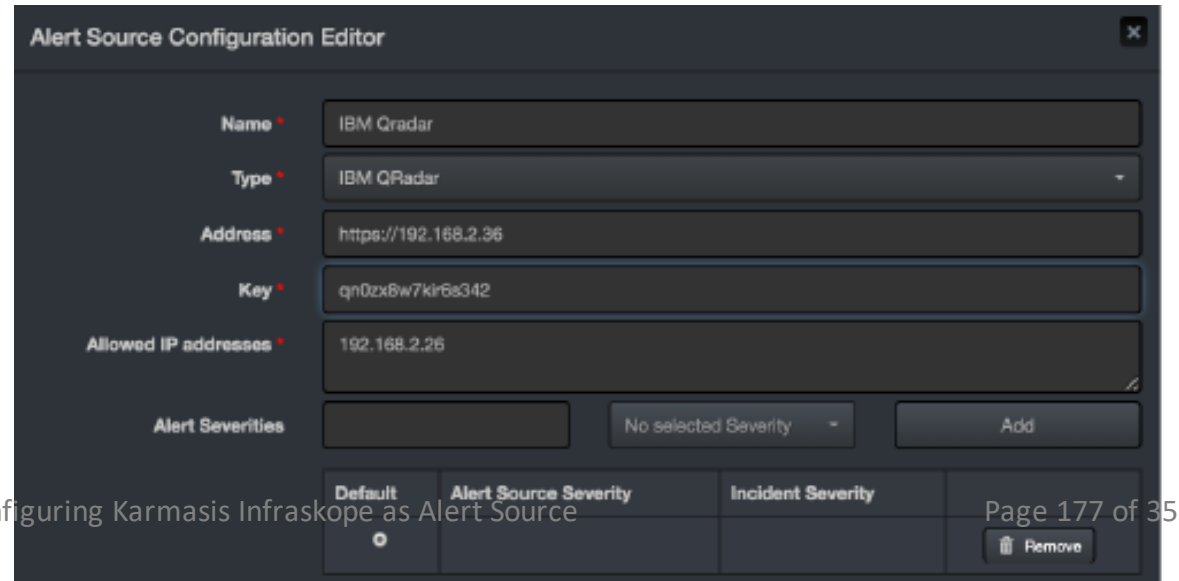
1. Click **Configuration** > **Alert Source** > **Create Alert Source Configuration**.
2. Specify the following parameter values in the **Configuration** form:

Parameter	Value
Name	Karmasis Infraskope Alert Source on SOAR
Type	Karmasis Infraskope
Address	Address of Karmasis Infraskope (in the following format:https://192.168.5.20)
Alert Severities	Mapping of alert severity values to SOAR incident severities

Configuration	Specify the following configuration parameters:
	`# Scope fields to be extracted from base events and/or correlated events (field1:CATEGORY:# CATEGORY is any of: EMAIL_ADDRESS, HASH, HOST, MAC_ADDRESS, NETWORK_ADDRESS,`
	`# COMPUTER_NAME, UNKNOWN, URL, USERNAME, PROCESS`
	`# ROLE is any of: OFFENDER, IMPACT, RELATED`
	`#`
	`# Note: The fields in the baseevent.scope and correlated.scope example below are always`
	`# Note: Extraction with same field name overrides the default one.`
	`# Note: Extraction with different field name does not override the default behaviour and`
	`# Note: Field names must start with / character`
	`#`
	`# Example: baseevent.scope=/targetusername:USERNAME: IMPACT`
	`# baseevent.scope=`
	`#`
	`# Example: correlated.scope=/AlertComputer:COMPUTER_NAME: OFFENDER`
	`# correlated.scope=`
	`# How far (in days) into the past ATAR will look for alerts at the initial sync task`
	`# If not provided, SOAR will use 14 days by default`
	`days.to.look.back.at.initial.sync=14`

Credential	Name of the credential set created on step 2 (For example, Karmasis Infraskope Credentials)
Trust Invalid SSL Certificates	Select this if Web UI's certificate certificate is self-signed is or is not recognized by browsers
Visible Alert Fields	Define the alarm fields to be displayed on Incident Management Service Desk


Alert Source Configuration Editor
Name *
Karmasis Infraskope
Type *
Karmasis Infraskope
Address *
https://192.168.200.84:8443
Alert Severities
No selected Severity
Add
Default
Alert Source Severity
Incident Severity
Service Unavailable
Urgent
Remove
Error
Critical
Remove
Configuration Content
Scope fields to be extracted from base events and/or correlated events (field1:CATEGORY:ROLE, field2:CATEGORY:ROLE, ...)
CATEGORY is any of: EMAIL_ADDRESS, HASH, HOST, MAC_ADDRESS, NETWORK_ADDRESS,
COMPUTER_NAME, UNKNOWN, URL, USERNAME, PROCESS
ROLE is any of: OFFENDER, IMPACT, RELATED
#
Note: The fields in the baseevent.scope and correlated.scope example below are always extracted by default.
Note: Extraction with same field name overrides the default one.
Credential *
Karmasis Infraskope Credentials
Create
Visible Alert Fields
Field Name
Visible Name
+ Add
Field Name
Visible Name
Actions
(No Data)
Total 0 ,
5
items / page
Trust Invalid SSL Certificates
Test
Close
Save


Alert Source Configuration Editor
Name *
IBM Qradar
Type *
IBM QRadar
Address *
https://192.168.2.36
Key *
qn0zx8w7klr6s342
Allowed IP addresses *
192.168.2.26
Alert Severities
No selected Severity
Add
Default
Alert Source Severity
Incident Severity
Remove

3. Click **Test** to test the integration.
4. Click **Save** to complete the integration.
5. If you have not configured `incident.autoSync=true` and want to forward offenses to SOAR from QRadar, navigate to **Configuration** > **Parameters**.
6. Select the value of **QRadarListenerEnabled** to true.

Integration Guide for Kaspersky Security Center

Integration Overview

ArcSight SOAR is capable of communicating with Kaspersky Security Center through WinRM and Powershell to block hashes, add tags to hosts, run tasks, move hosts to groups and retrieve information about various management objects.

Integration Capabilities

- Block (blacklist) SHA-256 or MD5 hash, with rollback support
- Add tag to host, with rollback support
- Move host to group
- Run task
- Retrieve host information

Configuration

Configuration on Kaspersky Security Center

- To define a Kaspersky Security Center installation as an integration on your

SOAR, following integration specific configuration should be performed.

- SOAR should be able to access the server with Kaspersky Security Center through WinRM on the network; usually with TCP port 5985 or 5986 (if SSL is enabled on WinRM). See WinRM Integration Guide for details on how to configure WinRM access.
- A local or domain administrator user account is required execute various capabilities.
- 32-bit version of Windows Scripting Host (which is available on a default Windows installation) is required to execute built-in scripts, which is usually located at C:\Windows\SYSWOW64\cscript.exe.

Configuring SOAR

- While creating this integration via Integrations tab of Configuration menu:
- Name: Display name of the integration.

- Address: Address of the integration. Format of the address should be

IP, IP:port, dns.hostname.localnet, or dns.hostname.localnet:port for HTTP;

or prefixed with https:// if HTTPS/SSL listener was enabled on WinRM.

- Credential: Credential that has been defined for this integration under the Credentials

menu.

Optional configuration

- `blockhash.categoryname:` Category name to add block hashes into; if unspecified defaults to SOAR. If specified category name doesn't exist, it will be automatically created.
- `path.cscriptexe:` Location of the 32-bits version of the cscript.exe on server.

If unspecified, defaults to "C:\\Windows\\SysWOW64\\cscript.exe".

Note: The backslashes must be escaped and double-backslash is required.

Overriding built-in scripts

SOAR allows overriding built-in scripts using Customization Library. Create a new customization of **Basic plugin script**, take note of its ID, and set the value of the script you'd like to override in the integration configuration by specifying its identifier as specified below:

Parameter Name	Description
enrichment.gettasknames	Retrieve names of tasks available for Run task capability
enrichment.getgroupnames	Retrieve names of groups available for Move host to group capability
enrichment.gettagnames	Retrieve names of tags available for Add tag to host capability
enrichment.hostinfo	Retrieve host information enrichment script
execute.blockhash	Block hash capability
rollback.blockhash	Rollback block hash capability
execute.addtag	Add tag capability
rollback.addtag	Rollback add tag capability
execute.movesystem	Move host to group capability
execute.runtask	Run task capability

Important points

- When these parameters are specified, built-in scripts will be ignored and the customization with specified ID will be used instead as the script. All scripts should target Windows Scripting Host with Javascript language, unless a different/compatible interpreter is specified in path.cscriptexe parameter in integration configuration. See [https://support.kaspersky.com/9291](Kaspersky Enterprise Security Administration Kit Automation10) for reference on using its COM/ActiveX API.
- SOAR's implementation is sensitive to the expected output of these scripts; overriding a capability with a script that doesn't write expected output to stdout may break existing functionality.
- Scripts are automatically evaluated as StringTemplate and various parameters are injected into the template for block hash, run task, move host into group, add tag and host information capabilities. See built-in scripts below for example usage and [http://www.stringtemplate.org](String Template Website) for more details on how to make use of the ST engine.

Example:

```
# 4214 is the ID of the customization to override this capability.
execute.runtask=4214
```

Built-in Tasks

Get Task Names

```
# Scope
function obj(name) {
return new ActiveXObject("klakaut.KlAk" + name);
}
try {
var oConnectProps = obj("Params"), oAdmServer = obj("Proxy"),
oSrvView = obj("SrvView"),
oTasks = obj("Tasks2"), item, enumObj;
oConnectProps.Add("Address", "127.0.0.1:13291");
```

```
oAdmServer.Connect(oConnectProps);
oTasks.AdmServer = oSrvView.AdmServer = oAdmServer;
enumObj = new Enumerator(oTasks.EnumTasks(-1));
WScript.Echo('[OK] [BEGIN]');
for (; !enumObj.atEnd(); enumObj.moveNext()) {
item = enumObj.item();
WScript.Echo(item.item('TASK_UNIQUE_ID') + '=' + item.item('DisplayName'));
}
WScript.Echo('[END]');
} catch (e) {
WScript.Echo("[Error] " + e.number + " occured !!! " + e.description);
}
```

Get Group Names

```
function obj(name) {
return new ActiveXObject("klakaut.KlAk" + name);
}
function EnumerateGroups(oSubgroupsEnum) {
var enumObj = new Enumerator(oSubgroupsEnum);
for (;!enumObj.atEnd();enumObj.moveNext()) {
var oObj = enumObj.item();
WScript.Echo(oObj.Item("id") + '=' + oObj.Item("name"));
if (oObj.Check("groups")) {
EnumerateGroups(oObj.Item("groups"));
}
}
}
try {
```

```
var oConnectProps = obj("Params"), oAdmServer = obj("Proxy"),
oGroups = obj("Groups");
oConnectProps.Add("Address", "127.0.0.1:13291");
oAdmServer.Connect(oConnectProps);
oGroups.AdmServer = oAdmServer;
WScript.Echo('[OK] [BEGIN]');
EnumerateGroups(oGroups.GetSubgroups(oGroups.GroupIdGroups, 0));
WScript.Echo('[END]');
} catch (e) {
WScript.Echo("[Error] " + e.number + " occured !!! " + e.description);
}
```

Get Tag Names

```
function obj(name) {
return new ActiveXObject("klakaut.KlAk" + name);
}
try {
var oConnectProps = obj("Params"), oAdmServer = obj("Proxy"),
oTagsControl = obj("TagsControl"), oProps = obj("Params"), oTags, enumObj;
oConnectProps.Add("Address", "127.0.0.1:13291");
oAdmServer.Connect(oConnectProps);
oTagsControl.AdmServer = oAdmServer;
oTagsControl.Prop("ListName") = "HostsTags";
oTags = oTagsControl.GetAllTags(oProps);
WScript.Echo('[OK] [BEGIN]');
if (oTags != null) {
enumObj = new Enumerator(oTags);
for (; !enumObj.atEnd(); enumObj.moveNext()) {
```

```
WScript.Echo(enumObj.item() + "=" + enumObj.item());
}
}
WScript.Echo('[END]');
} catch (e) {
WScript.Echo("[Error] " + e.number + " occured !!! " + e.description);
}
```

Host Information Enrichment

```
function obj(name) {
return new ActiveXObject("klakaut.KlAk" + name);
}
function ip2long(IPaddress) {
var ip = IPaddress.match(/^(\d+)\.(\d+)\.(\d+)\.(\d+)$/);
return ip ? (+ip[1] << 24) + (+ip[2] << 16) + (+ip[3] << 8) + (+ip[4]) : null;
}
function long2ip(l) {
return ((l >> 24) & 255) + "." + ((l >> 16) & 255) + "." + ((l >> 8) & 255) +
"." + (l & 255);
}
function coll() {
var ret = obj("Collection"), len = arguments.length, args = arguments;
if (len == 1) {
args = arguments[0].split('|');
len = args.length;
}
ret.SetSize(len);
for (var i=0; i<len; i++) {
```

```
ret.SetAt(i, (arguments.length == 1 ? "KLHST_WKS_" : "") + args[i]);
}
return ret;
}
function g(a, e) {
var r = e.item('KLHST_WKS_' + a);
if (r === undefined) {
r = '';
}
return r;
}
var rtpState = ["Unknown", "Stopped", "Suspended", "Starting", "Running",
"Running (Maximum protection)", "Running (Maximum speed)",
"Running (Recommended settings)", "Running (Custom settings)", "Failure"];
function getStatus(v) {
var r = [];
if ((v & 1) == 1) {
r.push("Visible");
}
if ((v & 4) == 4) {
r.push("Agent:Installed");
}
if ((v & 8) == 8) {
r.push("Agent:Alive");
}
if ((v & 16) == 16) {
r.push("Real-Time-Protection:Installed");
}
```

```
return r.join(",");
}
try {
var oConnectProps = obj("Params"), oAdmServer = obj("Proxy"),
oHosts = obj("Hosts"), c=0;
oConnectProps.Add("Address", "127.0.0.1:13291");
oAdmServer.Connect(oConnectProps);
oHosts.AdmServer = oAdmServer;
var fieldsToReturn = "LAST_VISIBLE|STATUS|RTP_STATE|LAST_UPDATE|LAST_FULLSCAN|
WINHOSTNAME|WINDOMAIN|OS_NAME|OS_VER_MAJOR|OS_VER_MINOR|IP_
LONG|PRODUCT_TAG_NAME";
var ftr = fieldsToReturn.split('|');
var enumObj = new Enumerator(oHosts.FindHosts("(KLHST_WKS_IP_LONG=" +
ip2long('%host%') + ")", coll(fieldsToReturn), coll()));
WScript.Echo('[OK]');
for (; !enumObj.atEnd(); enumObj.moveNext()) {
var e = enumObj.item();
WScript.Echo('[' + c++ + ']' +
'LAST_VISIBLE=' + Date.parse(g('LAST_VISIBLE', e)) +
'|LAST_UPDATE=' + Date.parse(g('LAST_UPDATE', e)) +
'|LAST_FULLSCAN=' + Date.parse(g('LAST_FULLSCAN', e)) +
'|WINHOSTNAME=' + g('WINHOSTNAME', e) +
'|WINDOMAIN=' + g('WINDOMAIN', e) +
'|OS=' + g('OS_NAME', e) + ' (' + g('OS_VER_MAJOR', e) + '.' +
g('OS_VER_MINOR', e) + ')' +
'|IP=' + long2ip(g('IP_LONG', e)) +
'|RTP_STATE=' + rtpState[g('RTP_STATE', e)] +
'|STATUS=' + getStatus(g('STATUS', e)) +
```

```
'|PRODUCT_TAG_NAME=' + g('PRODUCT_TAG_NAME', e)
);
}
WScript.Echo("[END] Retrieved information for " + c + " hosts.");
} catch (e) {
WScript.Echo("[Error] " + e.number + " occured !!! " + e.description);
}
```

Block Hash Action Capability

```
var hashes = [%hashes: {h | "%h%"}; separator=", "%];
function obj(name) {
return new ActiveXObject("klakaut.KlAk" + name);
}
try {
var oConnectProps = obj("Params"), oAdmServer = obj("Proxy"),
oCategory = obj("FileCategorizer"), oFields2Return = obj("Collection"),
oSrvView = obj("SrvView");
oConnectProps.Add("Address", "127.0.0.1:13291");
oAdmServer.Connect(oConnectProps);
oCategory.AdmServer = oSrvView.AdmServer = oAdmServer;
oFields2Return.SetSize(2);
oFields2Return.SetAt(0, "id");
oFields2Return.SetAt(1, "name");
var enumObj = new Enumerator(oSrvView.GetChunkAccessor('customcategories',
'(name = "*")', oFields2Return, obj("Collection"))), catFound = null;
for (; !enumObj.atEnd(); enumObj.moveNext()) {
var item = enumObj.item();
if (item.item('name') === '%categoryname%') {
```

```
catFound = item.item('id');
}
// dump("", "", item, false);
// dump("", "", oCategory.GetCategory(item.item('id')), false);
}
var oCatToAdd, oInclProps, i, oCatProps = obj("Params"), oCatData = catFound ?
oCategory.getCategory(catFound) : null, oInclusions = catFound ?
oCatData.Item('inclusions') : obj("Collection");
for (i=0; i<hashes.length; i++) {
oInclProps = obj("Params");
oInclProps.Add('ex_type', 3);
oInclProps.Add(hashes[i].length == 32 ? 'str' : 'str2', hashes[i]);
oInclProps.Add('str_op', 0);
oInclusions.SetSize(oInclusions.Count + 1);
oInclusions.setAt(oInclusions.Count - 1, oInclProps);
}
if (!catFound) {
oCatProps.Add('name', '%categoryname%');
oCatProps.Add('CategoryType', 0);
oCatProps.Add('inclusions', oInclusions);
oCatToAdd = oCategory.CreateCategory(oCatProps);
WScript.Echo("[OK] [CREATED] Added " + hashes.length +
' hashes to newly created category: %categoryname%');
} else {
oCategory.UpdateCategory(catFound, oCatData);
WScript.Echo("[OK] [UPDATED] Added " + hashes.length +
' hashes to existing category: %categoryname% its current size is: '
+ oInclusions.Count);
```

```
}
} catch (e) {
WScript.Echo("[Error] " + e.number + " occured !!! " + e.description);
}
```

Rollback of block hash capability

```
var hashes = [%hashes: {h | "%h%"}; separator=", "%];
function obj(name) {
return new ActiveXObject("klakaut.KlAk" + name);
}
try {
var oConnectProps = obj("Params"), oAdmServer = obj("Proxy"),
oCategory = obj("FileCategorizer"), oFields2Return = obj("Collection"),
oSrvView = obj("SrvView");
oConnectProps.Add("Address", "127.0.0.1:13291");
oAdmServer.Connect(oConnectProps);
oCategory.AdmServer = oSrvView.AdmServer = oAdmServer;
oFields2Return.SetSize(2);
oFields2Return.SetAt(0, "id");
oFields2Return.SetAt(1, "name");
var enumObj = new Enumerator(oSrvView.GetChunkAccessor('customcategories',
'(name = "*")', oFields2Return, obj("Collection"))), catFound = null;
for (; !enumObj.atEnd(); enumObj.moveNext()) {
var item = enumObj.item();
if (item.item('name') === '%categoryname%') {
catFound = item.item('id');
}
}
```

```
if (!catFound) {
WScript.Echo("[OK] [DOESNTEXIST] Category %categoryname% doesn't exist,
no need to remove anything.");
} else {
var oCatData = oCategory.getCategory(catFound),
oInclusions = oCatData.Item('inclusions'),
oNewInclusions = obj("Collection"), i, j, k=0;
for (j=0; j<oInclusions.Count; j++) {
for (i=0; i<hashes.length; i++) {
var incl = oInclusions.Item(j);
if (incl.Item('str') !== hashes[i] && incl.Item('str2') !== hashes[i]) {
oNewInclusions.SetSize(oNewInclusions.Count + 1);
oNewInclusions.setAt(oNewInclusions.Count - 1, incl);
} else {
k++;
}
}
}
oCatData.Item('inclusions') = oNewInclusions;
oCategory.UpdateCategory(catFound, oCatData);
WScript.Echo("[OK] [UPDATED] Removed " + k + " of " + hashes.length +
' hashes from category: %categoryname% its current size is: ' +
oNewInclusions.Count);
}
} catch (e) {
WScript.Echo("[Error] " + e.number + " occured !!! " + e.description);
}
```

Add tag to host capability

```
function obj(name) {
return new ActiveXObject("klakaut.KlAk" + name);
}
function ip2long(IPaddress) {
var ip = IPaddress.match(/^(\d+)\.(\d+)\.(\d+)\.(\d+)$/);
return ip ? (+ip[1] << 24) + (+ip[2] << 16) + (+ip[3] << 8) + (+ip[4]) : null;
}
try {
var oConnectProps = obj("Params"), oAdmServer = obj("Proxy"),
oTagsControl = obj("TagsControl"), oHosts = obj("Hosts"),
oFields2Return = obj("Collection"), enumObj, taggedHosts = 0;
oConnectProps.Add("Address", "127.0.0.1:13291");
oAdmServer.Connect(oConnectProps);
oTagsControl.Prop("ListName") = "HostsTags";
oTagsControl.AdmServer = oHosts.AdmServer = oAdmServer;
oFields2Return.SetSize(1);
oFields2Return.SetAt(0, "KLHST_WKS_HOSTNAME");
enumObj = new Enumerator(oHosts.FindHosts("(KLHST_WKS_IP_LONG=" + ip2long('%host%') +
")", oFields2Return, obj("Collection")));
for (; !enumObj.atEnd(); enumObj.moveNext()) {
var oTagArrayItem = obj("Params");
oTagArrayItem.Add("KLTAGS_VALUE", "%tag%");
oTagArrayItem.Add("KLTAGS_SET", true);
var oTagArray = obj("Collection");
oTagArray.SetSize(1);
oTagArray.SetAt(0, oTagArrayItem);
var oHostsArrayItem = obj("Params");
```

```
oHostsArrayItem.Add("KLTAGS_ITEM_ID", enumObj.item().item('KLHST_WKS_HOSTNAME'));
oHostsArrayItem.Add("KLTAGS_TAGS", oTagArray);
var oHostsArray = obj("Collection");
oHostsArray.SetSize(1);
oHostsArray.SetAt(0, oHostsArrayItem);
var oSetTagsCallProps = obj("Params");
oSetTagsCallProps.Add("KLTAGS_FULL_REPLACE", false);
oTagsControl.SetTags(oHostsArray, oSetTagsCallProps);
taggedHosts++;
}
WScript.Echo("[OK] Added '%tag%' to " + taggedHosts + " hosts.");
} catch (e) {
WScript.Echo("[Error] " + e.number + " occured !!! " + e.description);
}
```

Rollback of Add Tag to Host Capability

```
function obj(name) {
return new ActiveXObject("klakaut.KlAk" + name);
}
function ip2long(IPaddress) {
var ip = IPaddress.match(/^(\d+)\.(\d+)\.(\d+)\.(\d+)$/);
return ip ? (+ip[1] << 24) + (+ip[2] << 16) + (+ip[3] << 8) + (+ip[4]) : null;
}
try {
var oConnectProps = obj("Params"), oAdmServer = obj("Proxy"),
oTagsControl = obj("TagsControl"), oHosts = obj("Hosts"),
oFields2Return = obj("Collection"), enumObj, tagRemovedHosts = 0, removedTagCount;
oConnectProps.Add("Address", "127.0.0.1:13291");
```

```
oAdmServer.Connect(oConnectProps);
oTagsControl.Prop("ListName") = "HostsTags";
oTagsControl.AdmServer = oHosts.AdmServer = oAdmServer;
oFields2Return.SetSize(1);
oFields2Return.SetAt(0, "KLHST_WKS_HOSTNAME");
enumObj = new Enumerator(oHosts.FindHosts("(KLHST_WKS_IP_LONG=" +
ip2long('%host%') + ")", oFields2Return, obj("Collection")));
for (; !enumObj.atEnd(); enumObj.moveNext()) {
var hostId = enumObj.item().item('KLHST_WKS_HOSTNAME');
var oHostIds = obj("Collection");
oHostIds.setSize(1);
oHostIds.SetAt(0, hostId);
var oExistingTagArray = oTagsControl.GetTags(oHostIds, obj("Params"));
var oTagArray = obj("Collection");
removedTagCount = 0;
for (var i = 0; i < oExistingTagArray.Count; i++) {
var oTagEntry = oExistingTagArray.Item(i);
var oTagValues = oTagEntry.Item("KLTAGS_TAGS");
for (var j = 0; j < oTagValues.Count; j++) {
var tag = oTagValues.Item(j);
if (tag != '%tag%') {
oTagArray.SetSize(oTagArray.Count + 1);
var oTagArrayItem = obj("Params");
oTagArrayItem.Add("KLTAGS_VALUE", tag);
oTagArrayItem.Add("KLTAGS_SET", true);
oTagArray.SetAt(oTagArray.Count - 1, oTagArrayItem);
} else {
removedTagCount++;
```

```
}
}
}
var oHostsArrayItem = obj("Params");
oHostsArrayItem.Add("KLTAGS_ITEM_ID", hostId);
oHostsArrayItem.Add("KLTAGS_TAGS", oTagArray);
var oHostsArray = obj("Collection");
oHostsArray.SetSize(1);
oHostsArray.SetAt(0, oHostsArrayItem);
var oSetTagsCallProps = obj("Params");
oSetTagsCallProps.Add("KLTAGS_FULL_REPLACE", true);
oTagsControl.SetTags(oHostsArray, oSetTagsCallProps);
if (removedTagCount > 0) {
tagRemovedHosts++;
}
}
WScript.Echo("[OK] Removed '%tag%' from " + tagRemovedHosts + " hosts.");
} catch (e) {
WScript.Echo("[Error] " + e.number + " occured !!! " + e.description);
}
```

Move system to group capability

```
function obj(name) {
return new ActiveXObject("klakaut.KlAk" + name);
}
function ip2long(IPaddress) {
var ip = IPaddress.match(/^(\d+)\.(\d+)\.(\d+)\.(\d+)$/);
return ip ? (+ip[1] << 24) + (+ip[2] << 16) + (+ip[3] << 8) + (+ip[4]) : null;
```

```
}
try {
var oConnectProps = obj("Params"), oAdmServer = obj("Proxy"),
oHosts = obj("Hosts"), oFields2Return = obj("Collection"), enumObj,
hostsToMove = obj("Collection");
oConnectProps.Add("Address", "127.0.0.1:13291");
oAdmServer.Connect(oConnectProps);
oHosts.AdmServer = oAdmServer;
oFields2Return.SetSize(1);
oFields2Return.SetAt(0, "KLHST_WKS_HOSTNAME");
enumObj = new Enumerator(oHosts.FindHosts("(KLHST_WKS_IP_LONG=" +
ip2long('%host%') + ")", oFields2Return, obj("Collection")));
for (; !enumObj.atEnd(); enumObj.moveNext()) {
hostsToMove.SetSize(hostsToMove.Count + 1);
hostsToMove.SetAt(hostsToMove.Count - 1,
enumObj.item().item('KLHST_WKS_HOSTNAME'));
}
oHosts.MoveHostsToGroup(parseInt('%group%'), hostsToMove);
WScript.Echo("[OK] " + hostsToMove.Count + " hosts moved to group #%group%");
} catch (e) {
WScript.Echo("[Error] " + e.number + " occured !!! " + e.description);
}
```

Run task capability

```
function obj(name) {
return new ActiveXObject("klakaut.KlAk" + name);
}
try {
```

```
var oConnectProps = obj("Params"), oAdmServer = obj("Proxy"),
oTasks = obj("Tasks2"), item, enumObj, taskFound=false;
oConnectProps.Add("Address", "127.0.0.1:13291");
oAdmServer.Connect(oConnectProps);
oTasks.AdmServer = oAdmServer;
enumObj = new Enumerator(oTasks.EnumTasks(-1));
for (; !enumObj.atEnd(); enumObj.moveNext()) {
item = enumObj.item();
if (item.item('TASK_UNIQUE_ID') == '%task%') {
oTask = oTasks.GetTask(parseInt('%task%'));
oTasks.RunTask(parseInt('%task%'));
taskFound = oTask;
}
}
WScript.Echo(taskFound ? '[OK] Task #%task%:' + taskFound.item('DisplayName') +
' successfully started.' : '[ERROR] Specified task #%task% was not found.');
} catch (e) {
WScript.Echo("[Error] " + e.number + " occured !!! " + e.description);
}
```

Integration Guide for LogRhythm SIEM

Integration Overview

LogRhythm is an enterprise security information and event management (SIEM) solution for collecting, correlating, and reporting on security event information.

Integration Capabilities

ArcSight SOAR has the following integration capabilities with LogRhythm SIEM:

- Ingest Correlated Alerts
- Retrieve Base Events
- Execute Log Query
- Add to List
- Remove from List
- Add Alarm Comments
- Update Alarm Status
- Get Host Status
- Get Alarm Events by ID
- Get Alarm History by ID
- Get Alarm by ID

Use Case #1: Investigating and Mitigating Cyber-attacks

Enterprises get hundreds of security alerts every day, and SOC teams drown in the tsunami of alerts while trying to evaluate and prioritize those alerts. ATAR is integrated with LogRhythm SIEM to help with the prioritization and investigation of alerts as well as the remediation of incidents. When an alert comes a new incident is created on ATAR's own Incident Management Service Desk. ATAR then allow analysts to investigate the case and take remedial actions.

Configuration

Prerequisites

Currently SOAR supports LogRhythm SIEM version *7.4.6.

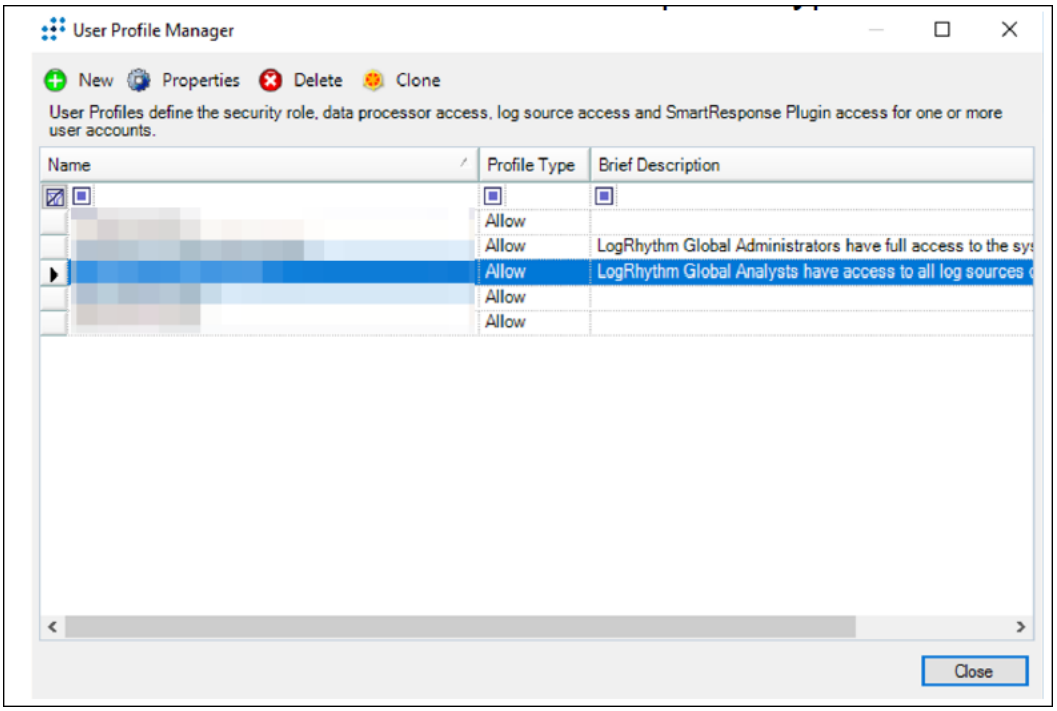
LogRhythm SIEM has both SOAP webservice and REST API with different capabilities.

SOAR connects to both LogRhythm SIEM SOAP Web service and REST API via HTTPS.By default SOAP webservice interface works on 443/tcp and REST-API interface works on 8501/tcp port. So access permission to these ports is required.

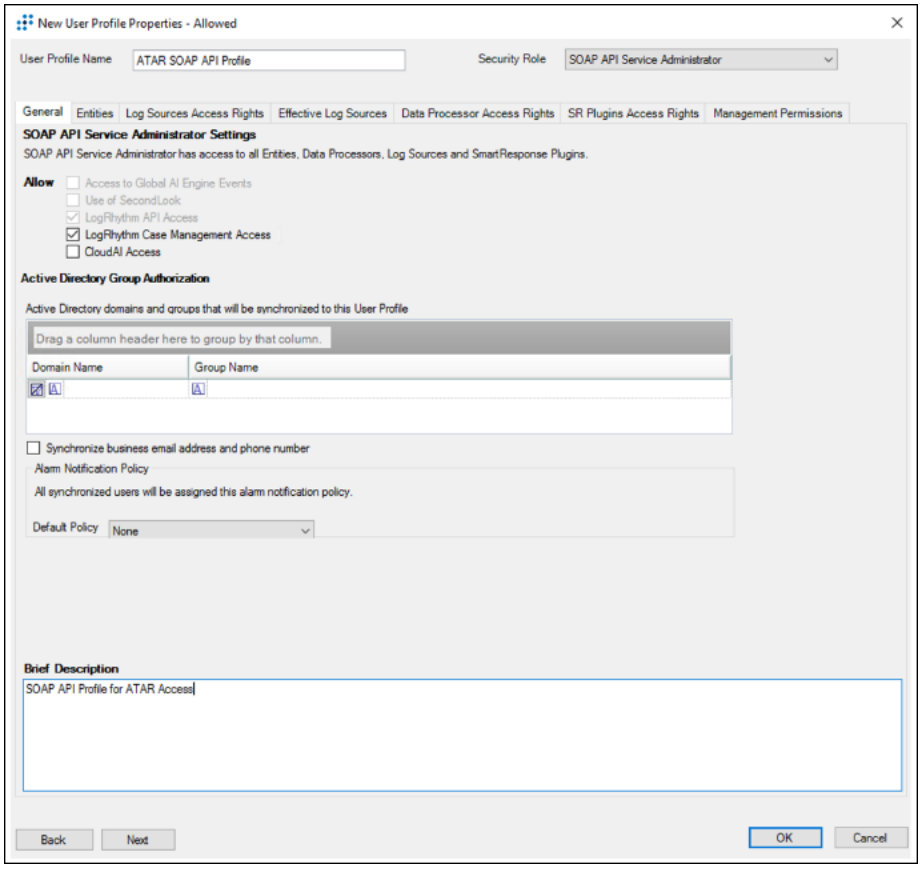
A user account and a Token for ATAR to connect to LogRhythm SIEM webservice and API.

Configuration on LogRhythm SIEM

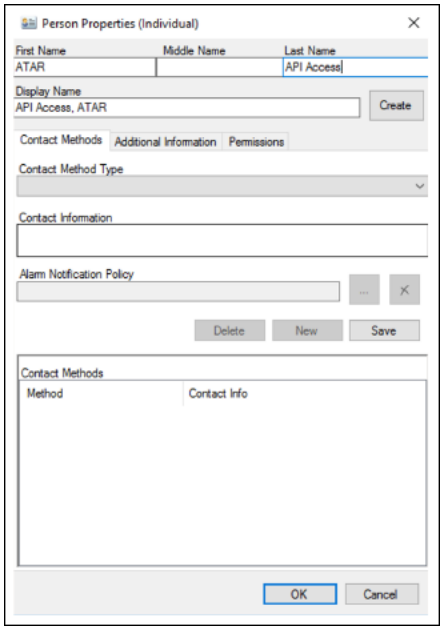
1. On LogRhythm Console application, navigate to **Tools** > **Administration** > **User Profile Manager** and create new Profile with **Allow Access** profile type.



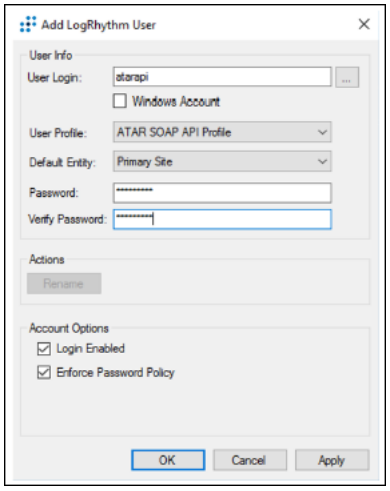
2. Select **SOAP API Service Administrator** as **Security Role**, and allow **LogRhythm Case Management Access**.



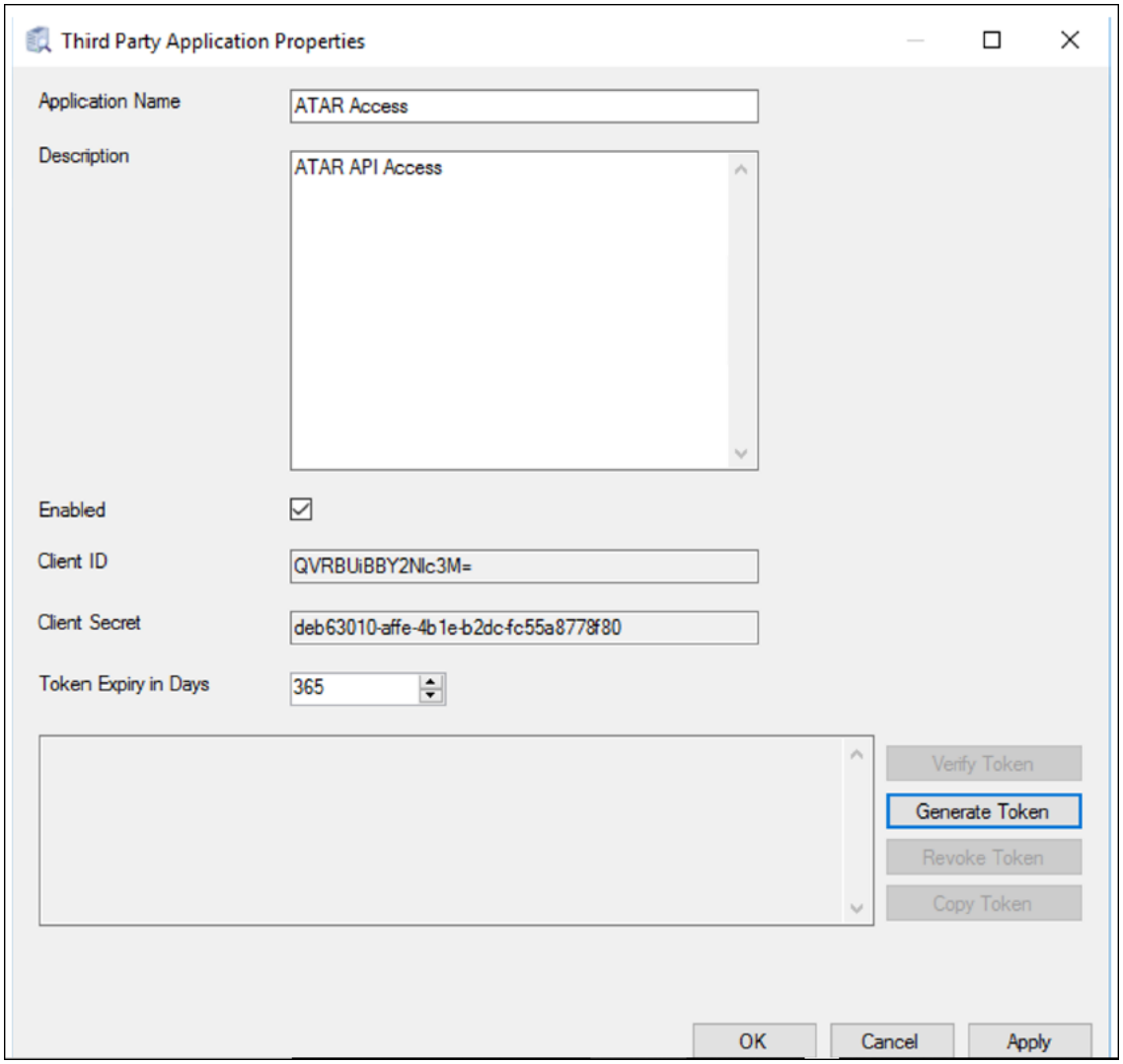
3. On **People** tab, create a new person (Individual Record Type) to be used for SOAR access:



4. Select the person record created on previous step and right-lick **Create User Account**:



5. On Third Party Application Properties tab, create a new record and generate a Token:



Configuration on ATAR

Configuring Credentials

1. Navigate to **Configuration** > **Credentials** and click **Create Credential**.
2. Fill the Credential Editor form as follows:
   a. Internal Credential:

      Type: Internal credential.

      Name: Display name of credential set (i.e., LogRhythm SIEM Credentials).

      Username: User you have created for ATAR on LogRhythm SIEM.

      Password: Password of the user you have created for ATAR on LogRhythm SIEM.

      Private Key: Third party application token you have created on LogRhythm SIEM.

   b. Credential Store:

      Type: External credential.

      Name: Name of the credential with pull path of the safe on store.

Configuring LogRhythm SIEM as Alert Source

1. Navigate to **Configuration** > **Alert Source** and click **Create Alert Source Configuration**.
2. Fill the configuration form as follows:

   Name: Display name of LogRhythm SIEM Alert Source on ATAR.

   Type: LogRhythmM.

   Address: Address of the LogRhythm SIEM Manager (the format should be https://192.168.5.20).

   Configuration: Specify the following configuration parameters.

```
# Scope fields to be extracted from base events and/or correlated events
(field1:CATEGORY:# CATEGORY is any of: EMAIL_ADDRESS, HASH, HOST, MAC_
ADDRESS, NETWORK_ADDRESS,

# COMPUTER_NAME, UNKNOWN, URL, USERNAME, PROCESS

# ROLE is any of: OFFENDER, IMPACT, RELATED

#
```

```
# Note: The fields in the baseevent.scope example below are always
extracted by default.

# Note: Extraction with same field name overrides the default one.

# Note: Extraction with different field name does not override the
default behaviour and # Note: Field names must start with / character

#

# Example: baseevent.scope=/account:USERNAME:IMPACT, /impactedIP:NETWORK_
ADDRESS:IMPACT, # baseevent.scope=

#

# Example: correlated.scope=/sourcev6:NETWORK_ADDRESS:OFFENDER

# correlated.scope=

# configure how far (in minutes) into the past this enrichment will look.

#cache.reusing.duration=20

# How far (in days) into the past ATAR will look for alerts at the
initial sync task

# If not provided, ATAR will use 14 days by default

days.to.look.back.at.initial.sync=14

#LogRhythm API timezone, if not specified GMT will be used as default

#timezone=GMT

# REST API port, keep empty for default configuration

rest.port=8501

# SOAP API port, keep empty for default configuration

soap.port=
```

Credential: Name of the credential set you've just created. (i.e., LogRhythm SIEM Credentials).

Trust Invalid SSL Certificates: Select this if server's certificate is self-signed or not recognized by browsers.

Visible Alert Fields: You may define which alarm fields will be displayed on Incident Management Service Desk.

3. When you click the **Test** button the following popup should be displayed if youra success message is displayed.

4. Click **Save** to complete integration.

Configuring LogRhythm SIEM as Integration

1. Navigate to **Configuration** > **Integrations** and click **Create Integration**.
2. Fill the configuration form as follows:

   Name: Display name of LogRhythm SIEM integration on ATAR.

   Type: LogRhythm SIEM.

   Address: Address of LogRhythm SIEM Manager (the format should be https://192.168.5.20).

   Configuration: You need to specify the following configuration parameters.

   ```
   #LogRhythm enrichment API timezone, if not specified GMT will be used as default
   ```

   ```
   #timezone=GMT
   ```

   ```
   # REST API port, keep empty for default configuration rest.port=8501
   ```

   ```
   # SOAP API port, keep empty for default configuration soap.port=
   ```

   Credential: Name of the credential set you've just created. (i.e., LogRhythm SIEM Credentials).

   Trust Invalid SSL Certificates: Select this if server's certificate is self-signed or not recognized by browsers.

   Require Approval From: Select user(s) from list to ask her/his approval before executing actions on this integration.

   Notify: Select user(s) from the list to notify when ATAR performs an action on this integration.
3. When you click the **Test** button a confirmation message is displayed
4. Click **Save** to complete integration.

Additional Notes

The following configuration parameters can be used for fine tuning the integration. Consult SOAR field engineering team before editing them:

Parameter Name	Description	Default Value
LogRhythmListenerMaxRetrySeconds	LogRhythm listener queue max message retry in seconds	1800
LogRhythmListenerQueueConcurrency	Upper limit of LogRhythm Listener consumer thread count	3
LogRhythmSyncPeriod	Period in seconds to sync LogRhythm incidents	60

Integration Guide for MAY Siber Scop NET

Integration Overview

MAY Siber Scop NET is a NAC platform that provides visibility to any connected device across the network by integrating switches, routers and firewalls.

Integration Capabilities

ArcSight SOAR has the following integration capability with MAY Siber Scop NET:

Block

Use Case: Isolating Mal-behaving PC

With MAY Siber Scop NET integration, while responding an incident ATAR may block malbehaving computers' network access in order to contain the attack and prevent further spread of the attack. Blocking the host can be performed automatically within a playbook or manually by an analyst.

Configuration

Prerequisites

- Currently SOAR supports MAY Siber Scop NET version 7.1.17.
- SOAR connects to MAY Siber Scop NET API via HTTPS. Typically it runs on 443/tcp port. So access to this service is required.
- An API key is required for SOAR to connect to MAY Siber Scop NET.

Configuring MAY Siber Scop NET

Login to MAY Siber Scop NET and create Web service key under **Settings** > **Global Settings** > **Web Service Key** menu.

Configuring SOAR

1. Navigate to **Configuration** > **Credentials** and click **Create Credential**.
2. Fill the Credential Editor form as follows:

a. **Internal Credential:**

Type: Internal credential.

Name: Display name of credential set (i.e., MAY Siber Scop NET Credential).

Username: Empty.

Password: Web Service Key you have created for ATAR on MAY Siber Scop NET.

Private Key: Empty.

b. **Credential Store:**

Type: External credential.

Name: Name of the credential with pull path of the safe on store.

3. Navigate to **Configuration** > **Integrations** and click **Create Integration**.
4. Fill the configuration form as follows:

   Name: Display name of MAY Siber Scop NET integration on SOAR.

   Type: MAY Siber Scop NET.

   Address: Address of the integration (the format should be https://1.1.1.1 or https://abc.example.com).

   Configuration: You need to specify the following configuration parameters.

```
# Blocked by message customization

# $incident. for incident, $rule. for rule ,$alert. for alert

# parameters can be used in reason

# $incident.serial$ for incident serial, $incident.subject$ for incident

# subject

# $rule.id$ for rule id, $rule.name$ for rule name

# for customize reasons followings can be uncomment

#block.reason=Blocked by ATAR - $incident.serial$ $incident.subject$

#rollback.reason=Rollbacked by ATAR - $incident.serial$
$incident.subject$
```

   Credential: Name of the credential set you've just created on step 2. (i.e., MAY Siber Scop NET Credential).

   Trust Invalid SSL Certificates: Select this if Engine's certificate is self-signed or not recognized by browsers. Not selected.

   Require Approval From: Select user(s) from list to ask her/his approval before executing actions on this integration.

Notify: Select user(s) from the list to notify when ATAR performs an action on this integration.

5. When you click the **Test** button a success message is displayed.
6. Click **Save** to complete integration.

Integration Guide for MAY Siber SIEM

MAY Siber SIEM interoparates with SOAR on mailing protocol (protocol name: incoming mail process).

Configuration

1. Create **Microsoft Exchange integration** via the **Integration** tab of Configuration menu.

   For more information, refer to Microsoft Exchange Integration Guide for details.
2. Credential must be the username and password that are defined for the created mailbox. After defining the integration, the ID number of the Microsoft Exchange integration in the Integrations list, must be copied to the Paramater value of IncomingMailDevice in Parameters tab of Configuration menu.
3. While creating this alert source via **Alert Source** tab of Configuration menu, specify the following:

   Name: Display name of the alert source.

   Type: MAY Siber SIEM.

   Allowed IP addresses: Addresses of the allowed IPs.
4. When you click the **Test** button a success message is displayed.
5. Click **Save** to complete alert source configuration.

Customization Library tab of Configuration menu:

_ID` of the **IncomingEmailProcessScript** must be copied to parameter value of the EmailScriptID in Parameters tab of Configuration menu.

Define the the mailbox for SOAR at MAY Siber SIEM.

In **IncomingEmailProcessScript**:

- var sender = "xxxxxx"; it must be MAY Siber SIEM’s email address
- var alertSourceName = "xxxx"; it must be the same name in Alert Source tab of Configuration menu.

Code the following script to IncomingEmailProcessScript in Custom Script tab of Configuration menu.

```
var sender = "xxxxxx"; '
```

```
var alertSourceName = "xxxx";
```

```
var scopeItemValues = [];
```

```
var scopeItemCategories = [];
var scopeItemRoles = [];
var mapType = {
cast: "CAST",
scopeItem: "SCOPEITEM"
}
var alert = {
ruleName: "",
details: {},
ticket: {
subject: "",
description: ""
}
};
var scopeItemRoleKeys = {
impact : "IMPACT",
offender : "OFFENDER"
}
var atarScopeItemKeys = {
mailKey : "EMAIL_ADDRESS",
hashKey : "HASH",
hostKey : "HOST",
macKey : "MAC_ADDRESS",
networkAddressKey : "NETWORK_ADDRESS",
computerNameKey : "COMPUTER_NAME",
unknownKey : "UNKNOWN",
urlKey : "URL",
usernameKey : "USERNAME",
```

```
fileDataKey : "FILEDATA"
}
var defaultKeyMap = [
{
type : mapType.cast,
key : "Aç#klama",
castTo : "alert.ticket.description",
},
{
type : mapType.cast,
key : "Kural Ad#",
castTo : "alert.ruleName", // If you remove this, default will be mail subject
},
{
type : mapType.cast,
key : "Konu",
castTo : "alert.ticket.subject", // If you remove this, default will be mail subject
},
{
type : mapType.scopeItem,
key : "SOURCE_IP",
mapTo : atarScopeItemKeys.networkAddressKey,
mapAs : scopeItemRoleKeys.impact
},
{
type : mapType.scopeItem,
key : "Sender Email",
```

```
mapTo : atarScopeItemKeys.mailKey,
mapAs : scopeItemRoleKeys.offender
},
{
type : mapType.scopeItem,
key : "SOURCE_ADDRESS",
mapTo : atarScopeItemKeys.networkAddressKey,
mapAs : scopeItemRoleKeys.offender
},
{
type : mapType.scopeItem,
key : "CLIENT IP",
mapTo : atarScopeItemKeys.networkAddressKey,
mapAs : scopeItemRoleKeys.impact
},
{
type : mapType.scopeItem,
key : "DESTINATION_HOSTNAME",
mapTo : atarScopeItemKeys.hostKey,
mapAs : scopeItemRoleKeys.impact
},
{
type : mapType.scopeItem,
key : "DESTINATION_IP",
mapTo : atarScopeItemKeys.networkAddressKey,
mapAs : scopeItemRoleKeys.impact
},
{
```

```
type : mapType.scopeItem,
key : "HOSTNAME",
mapTo : atarScopeItemKeys.hostKey,
mapAs : scopeItemRoleKeys.impact
},
{
type : mapType.scopeItem,
key : "USERNAME",
mapTo : atarScopeItemKeys.usernameKey,
mapAs : scopeItemRoleKeys.impact
}
]
var subjectBasedMapping = [
Atar v3.0 Page 4 of 5
{
subject: "test2",
keyMap: [
{
key : "HostName",
mapTo : atarScopeItemKeys.hostKey,
mapAs : scopeItemRoleKeys.offender
},
{
key : "Mail",
mapTo : atarScopeItemKeys.mailKey,
mapAs : scopeItemRoleKeys.impact
}
],
```

```
}
];
if (mailMessage.getFrom() == sender) {
alert.ruleName = mailMessage.getSubject();
alert.ticket.subject = mailMessage.getSubject();
var lines = mailMessage.getTextBody().trim().split('\n');
var scopeMappingObj = subjectBasedMapping.find(o=> o.subject ==
mailMessage.getSubject().trim());
if (scopeMappingObj !== undefined) {
//Prepare Keymaps
subjectBasedMapping[0].keyMap.forEach(function (mapElement, index) {
var found = false;
var a = defaultKeyMap.map(function (o) {
if (o.key === mapElement.key) {
found = true;
mapElement.type = mapType.scopeItem;
return mapElement;
} else {
return o;
}
});
};
defaultKeyMap = a;
if (found === false) {
defaultKeyMap.push(mapElement);
};
}
};
```

```
}
for (i in lines) {
var splitted = lines[i].split(':');
if (splitted.length < 2) {
continue;
}
var key = splitted[0];
splitted.shift();
var value = splitted.join(':');
var mapObj = defaultKeyMap.find(o => o.key === key);
if (mapObj !== undefined) {
if (mapObj.type === mapType.scopeItem) {
//it is a scope item
scopeItemValues.push(value);
scopeItemCategories.push(mapObj.mapTo);
scopeItemRoles.push(mapObj.mapAs);
} else if (mapObj.type === mapType.cast) {
Atar v3.0 Page 5 of 5
var evalStr = '(' + mapObj.castTo + ' = \'' + value + '\')';
eval(evalStr);
}
}
}
var params = JSON.stringify(alert);
atar.alert(atar.alertSource(alertSourceName), params, "1000", "5",
scopeItemCategories,
scopeItemRoles, scopeItemValues);
}
```

Integration Guide for McAfee Enterprise Security Manager (ESM)

Integration Overview

McAfee Enterprise Security Manager is a SIEM solution which includes threat intelligence feeds, correlation, analytics, profiling, security alerts, data presentation and compliance with core FEATURES OF collecting, correlating and reporting incidents within network and application, and provide log management capabilities.

Integration Capabilities

ArcSight SOAR has the following integration capabilities with McAfee Enterprise Security Manager:

- Ingest Alerts
- Retrieve Base Events

Use Case #1: Incident Case Management

Integrated with McAfee Enterprise Security Manager, SOAR helps SOC teams to validate and prioritize the alerts. When an alert comes a new incident is created on SOAR's own Incident Management Service Desk. SOAR then allow analysts to investigate the case and take remedial actions.

Configuration

Prerequisites

- Currently SOAR supports McAfee Enterprise Security Manager (ESM) version 10.3.0

20180508 and later.

- McAfee Enterprise Security Maanager sends correlated event data (alert) to the listener on sOAR. By default this listener works on 9095/tcp port. So network traffic from Arcsight ESM to SOAR for this port must be permitted.
- SOAR connects to McAfee Enterprise Security Manager API trough HTTPS. By default RESTAPI interface works on 443/tcp port. So access permission to this port is required.
- A user account for SOAR to connect to McAfee ESM API.

Configuration on McAfee Enterprise Security Manager (ESM)

1. Navigate to System **Properties** > **Users and Groups** menu and add a user as a memberof **API_Group**.
2. Navigate through System **Properties** > **Event Forwarding** menu and add a new forwarding destination with the following parameters:
   - **Device:** Correlation Engine
   - **Destination IP:** IP address of SOAR server
   - **Destination Port:** 9095 (default value on SOAR)
   - **Protocol:** TCP
3. Set log format to **Standard Event Format**.
4. Select **Time Zone**.
5. Click **Event filter.**

Configuring SOAR

1. Navigate to **Configuration** > **Credentials** and click **Create Credential**.
2. Fill the Credential Editor form as follows:

   a. Internal Credential:

   Type: Internal credential.

   Name: Display name of credential set (i.e., McAfee ESM Credentials).

   Username: User you have created for SOAR on McAfee ESM.

   Password: Password of the user you have created for SOAR on McAfee ESM.

   Private Key: Empty.

   b. **Credential Store:**

   Type: External credential.

   Name: Name of the credential with pull path of the safe on store.
3. Navigate to **Configuration** > Alert **Source** and click **Create Alert Source** Configuration.
4. Fill the configuration form as follows:

   Name: Display name of McAfee Enterprise Security Manager integration on SOAR.

   Type: McAfee Enterprise Security Manager.

   Address: Address of the integration (the format should be https://192.168.2.35).

   Key: SOAR McAfee ESM listener will discard any data without this key.

Allowed IP Addresses: IP address of the McAfee ESM server. SOAR McAfee ESM listener will discard any data if it's not coming this address.

Alert Severities: Mapping of alert severity values to SOAR incident severities.

Configuration: You need to specify the following configuration parameters.

configure how far (in minutes) into the past this enrichment will look.

#cache.reusing.duration=20

Credential: Name of the credential set you've just created on step 2. (i.e., Arcsight Logger Credentials).

Visible Alert Fields: You may define which alarm fields will be displayed on Incident Management Service Desk.

Trust Invalid SSL Certificates: Select this if Engine's certificate is self-signed or not recognized by browsers.

5. When you click the **Test** button a success message is displayed.
6. Click **Save** to complete integration.

Additional Notes

The following configuration parameters can be used for fine tuning the integration.

Consult SOAR field engineering team before editing them:

- Parameter Name Description Default Value
- McAfeeListenerEnabled Enable McAfee Listener false
- McAfeeListenerPort McAfee Listener port 9095

Integration Guide for McAfee ePolicy Orchestrator

Integration Overview

McAfee ePolicy Orchestrator (ePO) is a management server for McAfee products which are used to protect endpoints from malware and network threats. It provides a centralized management console to simplify and accelerate the security effectiveness with visibility and control from device to cloud.

Integration Capabilities

- SOAR has the following integration capabilities with McAfee ePolicy Orchestrator:
- Assign Policy
- Apply Tag
- Host Information
- Move Host
- Run Task
- Set TIE Reputation

Use Case: Examining suspicious endpoint

With this integration, during the investigation of an incident SOARmay start an on-demand scan on a suspicious endpoint and may force new policy or move host to other place in system tree regarding scan result. This can be performed automatically within a playbook or manually by an analyst.

Configuration

Prerequisites

- Currently SOAR supports McAfee ePolicy Orchestrator version 5.3.2.
- SOAR connects to McAfee ePolicy Orchestrator API through HTTPS. Typically it runs on 8443/tcp port. So access to this service is required.
- An user account is required for SOAR to connect McAfee ePolicy Orchestrator.

Configuration on McAfee ePolicy Orchestrator

1. Navigate to **User Management** > **Permission** Sets and create a permission set for SOAR with the following permissions:

   System Tree access: Can search on the following nodes and portions of the System

   Tree: My Organization and Can access the following nodes and portions of the

   System Tree: My Organization

   Systems: Edit System Tree groups and systems

   Queries and Reports: Use public groups

   McAfee Agent: View and change policy settings

   Endpoint Security Threat Prevention: View and change task settings

   McAfee TIE Reputations: View and change reputations (if you have TIE server managed by your ePolicy Orchestrator)
2. View and change policy settings for the products that you want SOAR to chance policies for (for example: Endpoint Security Threat Prevention, Endpoint Security Firewall, Active Response, etc.)
3. Navigate **User Management** > **Users** and create a user with permission set you in previous step.

Configuring SOAR

1. Navigate to **Configuration** > **Credentials** and click **Create Credential.**
2. Fill the Credential Editor form as follows:

   a. **Internal Credential:**

   Type: Internal credential.

   Name: Display name of credential set (i.e., McAfee ePO Credentials).

   Username: Username you have configured on McAfee ePolicy Orchestrator.

   Password: Password for the user you have configured on McAfee ePolicy Orchestrator.

   Private Key: Empty.

   b. **Credential Store:**

   Type: External credential.

   Name: Name of the credential with pull path of the safe on store.
3. Navigate to **Configuration** > **Integrations** and click **Create Integration**.
4. Fill the configuration form as follows:

Name: Display name of McAfee ePolicy Orchestrator integration on ATAR.

Type: McAfee ePolicy Orchestrator.

Address: Address of the integration (the format should be https://192.168.2.100:8443).

Configuration: You need to specify the following configuration parameters. For the first integration these values can be left as is:

```
system.move.autoSort=false
```

```
clienttask.run.retryAttempts =
```

```
clienttask.run.retryIntervalInSeconds =
```

```
clienttask.run.abortAfterMinutes =
```

```
clienttask.run.useAllAgentHandlers =
```

```
clienttask.run.stopAfterMinutes=
```

```
clienttask.run.randomizationInterval =
```

```
policy.assignToSystem.resetInheritance=
```

Credential: Name of the credential set you've just created on step 2. (i.e., McAfeeePO Credentials).

Trust Invalid SSL Certificates: Select this if Engine's certificate is self-signed ornot recognized by browsers.

Require Approval From: Select user(s) from list to ask her/his approval beforeexecuting actions on this integration.

Notify: Select user(s) from the list to notify when SOARperforms an action on thisintegration.

5. When you click the **Test** button a success message is displayed.
6. Click **Save** to complete integration.

Integration Guide for McAfee Network Security Platform (IPS)

Integration Overview

McAfee Network Security Platform is an intrusion prevention system (IPS) to identify malicious network traffic and stops never-before-seen attacks for which no signatures exist.

Integration Capabilities

SOAR has the following integration capabilities with McAfee Network Security Platform:

- Blacklist MD5 Hash
- Quarantine IP address

Configuration

Prerequisites

- Currently SOAR supports McAfee Network Security Platform version 9.2.7.22.
- SOAR connects to McAfee Network Security Platform's API via HTTPS. By default McAfee Network Security Platform REST-API interface works on 443/tcp port. So access permission to this port is required.
- A user account is required for SOAR to connect McAfee Network Security Platform.

Configuration on McAfee Network Security Platform (IPS)

1. Navigate to **Manager** > **Users and Roles** > **Users** and create a user account with Super User role. In order to access API, Super User role is needed.
2. Navigate to **Devices** and note the device/sensor names.

Configuring SOAR

1. Navigate **Configuration** > **Credentials** and click **Create Credential**.
2. Fill the Credential Editor form as follows:

a. Internal Credential:

Type: Internal credential.

Name: Display name of credential set (i.e., McAfee NSP Credentials).

Username: User you have created for SOAR on McAfee Network Security Platform.

Password: Password of the user you have created for SOAR on McAfee Network Security Platform.

Private Key: Empty.

b. Credential Store:

Type: External credential.

Name: Name of the credential with pull path of the safe on store.

3. Navigate to **Configuration** > **Integrations** and click **Create Integration**.
4. Fill the configuration form as follows:

   Name: Display name of McAfee Network Security Platform integration on SOAR.

   Type: McAfee Network Security Platform.

   Address: Address of the integration (the format should be https://192.168.2.2).

   Credential: Name of the credential set you've just created on step 2. (i.e., McAfee NSP Credentials).

   Trust Invalid SSL Certificates: Select this if Platform's certificate is self-signed or not recognized by browsers.

   Configuration: You need to specify the following configuration parameters.

   # Name of ISP Devices/Sensors. You may write multiple device names separated by '|' character.

   SENSOR_NAME=SENSOR1|SENSOR2

   #proxy.id=5442

   Require Approval From: Select user(s) from list to ask her/his approval before executing actions on this integration.

   Notify: Select user(s) from the list to notify when SOAR performs an action on this integration.
5. When you click on the **Test** button a success message is displayed.
6. Click **Save** to complete integration.

Integration Guide for McAfee Web Gateway

Integration Overview

McAfee Web Gateway is a web filtering solution which utilizes both reputation and categorybased filtering and protection against zero-day malware as well.

Integration Capabilities

SOAR has the following integration capability with McAfee Web Gateway:

- Block URL

Use Case: Blocking access to malicious URL

SOAR can integrate with McAfee Web Gateway to block malicious URLs detected while responding an incident. Blocking can be performed automatically within a playbook or manually by an analyst.

Configuration

Prerequisites

- Currently SOAR supports McAfee Web Gateway version 7.7.2.8.0.
- SOAR connects to McAfee Web Gateway's API through HTTPS. By default McAfee Web Gateway REST-API interface works on 4712/tcp port. So access permission to this port is required.
- A user account for SOAR to connect to McAfee Web Gateway.

Configuration on McAfee Web Gateway

1. Navigate to **Accounts** menu and add a new Role to be used for SOAR user. The new role should have at least "Rest-Interface Accessible" permission.
2. Navigate through Accounts menu and add an Internal Administrator Account with the role you have created in previous step.
3. Create a Wildcard Expression List under **Policy** > **Lists**.
4. Create a new rule and enable it under **Policy** > **Rule Sets** > **URL Filtering** menu to use list created in previous step. Rule criteria should be:

URL.Host matches in list ATARBlock

5. Save changes.

Configuration on SOAR

1. Navigate **Configuration** > **Credentials** and click **Create Credential**.
2. Fill the Credential Editor form as follows:

   a. Internal Credential:

   Type: Internal credential.

   Name: Display name of credential set (i.e., McAfee Web GW Credential).

   Username: User you have created for SOAR on McAfee Web Gateway.

   Password: Password of the user you have created for SOAR on McAfee Web Gateway.

   Private Key: Empty.

   b. Credential Store:

   Type: External credential.

   Name: Name of the credential with pull path of the safe on store.
3. Navigate **Configuration** > **Integrations** and click **Create Integration.**
4. Fill the configuration form as follows:

   Name: Display name of McAfee Web Gateway integration on SOAR.

   Type: McAfee Web Gateway.

   Address: Address of the integration (the format should be 192.168.1.1:4712).

   Configuration: You need to specify the following configuration parameters.

```
# Use the McAfee Web Gateway management interface to create the
# list in Policy -> Rule set -> URL filtering section. SOAR will use
# specified list name when adding blocked items.
block.list.name=ATARBlock
```

   Credential: Name of the credential set you've just created on step 2. (i.e., McAfeeWeb GW Credential).

   Trust Invalid SSL Certificates: Select this if Engine's certificate is self-signed ornot recognized by browsers.

   Require Approval From: Select user(s) from list to ask her/his approval before executing actions on this integration.

   Notify: Select user(s) from the list to notify when SOAR performs an action on this integration.

5. On Integration editor, click **Show Additional Parameters** checkbox and set **ConnectionLimit** to “1” . Because of a limitation of McAfee Web Gateway, this value should never be greater than “1”.
6. When you click the **Test** button the following popup should be displayed if yourcredential and address is valid.
7. Click **Save** to complete integration.

Integration Guide for Micro Focus Arcsight ESM

Integration Overview

Micro Focus ArcSight ESM is an enterprise security information and event management

(SIEM) solution for collecting, correlating, and reporting on security event information.

Integration Capabilities

SOAR has the following integration capabilities with Micro Focus Arcsight ESM:

- Ingest Correlated Alerts
- Retrieve Base Events
- Create Case
- Update Case
- Search Cases
- Get Case Details
- Query Active List
- Add Entries to Active List
- Delete Entries from Active List

Use Case #1: Investigating and Mitigating Cyber-attacks

Enterprises get hundreds of security alerts every day, and SOC teams drown in the tsunami of alerts while trying to evaluate and prioritize those alerts. SOAR is integrated with Micro Focus ArcSight to help with the prioritization and investigation of alerts as well as the remediation of incidents. When an alert comes a new incident is created on SOAR's own Incident Management Service Desk. SOAR then allow analysts to investigate the case and take remedial actions.

Configuration

Prerequisites

Micro Focus Arcsight ESM sends correlated event data (alert) to the listener on SOAR. By default this listener works on 9090/tcp port. So network traffic from Arcsight ESM to SOAR for this port should be permitted.

SOAR connects to Micro Focus Arcsight ESM API using HTTPS.

By default REST-API interface works on 8443/tcp port. So access permission to this port is required.

A user account for SOAR to connect to Micro Focus Arcsight ESM API.

Configuration on Micro Focus Arcsight ESM

If Super Connector is installed and configured on Arcsight ESM, you may skip to step 3.

Otherwisei you need to install and configure Super Connector on Arcsight ESM. First log

into Arcsight Console and select “Users” on top menu and create a user with Forwarding

Connector type

Please ask for your system administrator for installing Super Connector package on

Arcsight ESM. Then follow the steps below for configuration

$ cd /opt/arcsight/MicroFocus_ArcsightSmartConnectors/SuperConnector/current/bin (directory $ ./runagentsetup.sh

Assuming ARCSIGHT_HOME: /opt/arcsight/MicroFocus_ ArcsightSmartConnectors/SuperConnector/current

Assuming JAVA_HOME: /opt/arcsight/MicroFocus_ ArcsightSmartConnectors/SuperConnector/current/Need to add localconnector cert

Assuming ARCSIGHT_HOME: /opt/arcsight/MicroFocus_ ArcsightSmartConnectors/SuperConnector/current

Assuming JAVA_HOME: /opt/arcsight/MicroFocus_ ArcsightSmartConnectors/SuperConnector/current/ArcSight Connectors starting...

ArcSight Agent Setup starting...

Connector Setup Wizard starting in mode [CONSOLE]

[Fri Nov 29 04:23:02 EET 2019] [INFO] Checking for a running instance of connector...

[Fri Nov 29 04:23:13 EET 2019] [INFO] Starting up connector...

Connector Setup

What would you like to do?

0- Add a Connector <-----

1- Enable FIPS mode

Please select an option: [Add a Connector] [0..1/cancel] :0

Select the connector to configure

Type:

0- ArcSight Forwarding Connector (Enhanced) <-----

Please select an option [0..0]: 0

Please verify the following parameters

Type: ArcSight Forwarding Connector (Enhanced)

Are the values correct [yes/no/back/cancel]?yes

Enter the parameter details

****

WARNING: Some of the required parameters will contain security

sensitive information. Do you want to hide the input for this

parameters from the screen?[yes/no]

(typically you would answer 'NO' only if you are using a slow

link (like a serial RS232 or a very slow network link) since

this may add additional delays to the connection. If you are

not sure, then select 'YES' or hit enter.

****

[yes]?

Input for private parameters will be hidden.

ArcSight Source Manager Host Name[localhost]: 192.168.5.5

ArcSight Source Manager Port[8443]:

ArcSight Source Manager User Name: <----- Username you've created on Arcsight ESM Console

R

ArcSight Source Manager Password: <----- Password of the user you've created on Arcsight
Please verify the following parameters

ArcSight Source Manager Host Name: 192.168.5.5

ArcSight Source Manager Port: 8443

ArcSight Source Manager User Name: ********

ArcSight Source Manager Password: ********

Are the values correct [yes/no/back/cancel]?yes

Then configure Super connector to send correlated events to SOAR listener:

$ cd /opt/arcsight/MicroFocus_ArcsightSmartConnectors/SuperConnector/current/bin
(directory $ ./runagentsetup.sh

Assuming ARCSIGHT_HOME: /opt/arcsight/MicroFocus_
ArcsightSmartConnectors/SuperConnector/current

Assuming JAVA_HOME: /opt/arcsight/MicroFocus_
ArcsightSmartConnectors/SuperConnector/current/ArcSight Agent Setup starting...

Connector Setup Wizard starting in mode [CONSOLE]

[Fri Nov 29 02:45:29 EET 2019] [INFO] Checking for a running instance of connector...

Connector Setup

What would you like to do?

0- Modify Connector <-----

1- Uninstall as a service

2- Enable FIPS mode

3- I do not want to change any setting

Please select an option: [Modify Connector] [0..3/cancel] :0

What would you like to do with the connector?

0- Modify connector parameters

1- Add, modify, or remove destinations <-----

Please select an option: [Modify connector parameters] [0..1/back/cancel] :1

Modify an existing destination or add a new destination

0- Add destination <-----

Please select an option: [0/back/cancel] 0:

Enter the type of destination

0- ArcSight Manager (encrypted)

1- ArcSight Logger SmartMessage (encrypted)

2- NSP Device Poll Listener

3- CEF Syslog <-----

4- CSV File

5- HP Operations Manager

6- HP Operations Manager i

Please select an option: [ArcSight Manager (encrypted)] [0..6/back/cancel] :3

Enter these parameters

Ip/Host: 192.168.15.10

Port: 9090

Protocol:

0- UDP

1- Raw TCP <-----

2- TLS

Please select an option [0..2][UDP]: 1

Forwarder:

0- true

1- false <-----

Please select an option [0..1][false]: 1

Please verify the following parameters

Ip/Host: 192.168.15.10 <----- IP Address of SOAR instance

Port: 9090 <----- Arcsight listener port on SOAR

Protocol: Raw TCP

Forwarder: false

Are the values correct [yes/no/back/cancel]?yes

Log in to Arcsight Console and create a new List to refer rules you want to forward to SOAR.

Add the rule names to the list you've created on previous step

On Arcsight Console create a "Pre-persistence Rule" to process and forward alerts to SOAR as follows.

You need to select the forwarding user you've created as owner of this rule

You need to set Action for this rule to add a key value to event data before sending SOAR.

7. Create another user account with Web User type on Arcsight Console for SOAR's access to Arcsight ESM's REST API and make sure that this user has This user has permission to read "all potential base events triggering correlations".

Configuration on SOAR

Configuring Credentials

Navigate Configuration -> Credentials and click on Create Credential.

Fill the Credential Editor form as follows:

a. Internal Credential:

Type: Internal credential.

Name: Display name of credential set (i.e., Arcsight ESM Credentials).

Username: Web user you have created for SOAR on Micro Focus Arcsight ESM.

Password: Password of the user you have created for SOAR on Micro Focus Arcsight ESM.

Private Key: Empty.

b. Credential Store:

Type: External credential.

Name: Name of the credential with pull path of the safe on store.

Configuring Micro Focus Arcsight ESM as Alert Source

Navigate Configuration -> Alert Source and click on Create Alert Source Configuration

.

Fill the configuration form as follows:

Name: Display name of Micro Focus Arcsight ESM Alert Source on SOAR.

Type: Micro Focus Arcsight ESM.

Address: Address of the Micro Focus Arcsight ESM Manager (the format should be

https://192.168.5.5:8443).

Key: Key you've set in the Pre-persistence rule definition. SOAR Arcsight listener will

discard any data without this key.

`Allowed IP Addresses: IP address of the Arcsight ESM Manager. SOAR Arcsight

listener will discard any data if it's not coming this address.

`Alert Severities: Mapping of alert severity values to SOAR incident severities.

Configuration: You need to specify the following configuration parameters.

CEF field to be used as severity value when mapping

ArcSight severity value to SOAR incident severity.

#

Suggested values: priority, severity, flexString1, flexNumber1

severity.field=priority

CEF-extension field to be used as rule name value (if not set, the CEF-header name field #severity.field=

Scope fields to be extracted from correlated event (field1:CATEGORY:ROLE, field2:CATEGORY:# CATEGORY is any of: EMAIL_ADDRESS, HASH, HOST, MAC_ADDRESS, NETWORK_ADDRESS,

COMPUTER_NAME, UNKNOWN, URL, USERNAME, PROCESS

ROLE is any of: OFFENDER, IMPACT, RELATED

#

Note: src:NETWORK_ADDRESS:OFFENDER, dst:NETWORK_ADDRESS:IMPACT, request:URL:OFFENDER

fields are always extracted by default.

Note: this parameter can specify additional fields to be extracted, and will not override

Example: correlated.scope=s_user:USERNAME:OFFENDER, dvc:NETWORK_ADDRESS:RELATED

correlated.scope=src:NETWORK_ADDRESS:OFFENDER, dst:NETWORK_ADDRESS:IMPACT, request:URL:# Additional scope fields to be extracted from base events (field1:CATEGORY:ROLE, field2:# Field names use JSON pointer notation. See the correlated.scope property for CATEGORY

Note: this parameter can specify additional fields to be extracted, and will not override

Example: baseevent.scope=/device/address:NETWORK_ADDRESS:RELATED

baseevent.scope=

configure how far (in minutes) into the past this enrichment will look.

#cache.reusing.duration=20

Enable/disable base events activity in the incident timeline

#enable.baseevent.activity=false

Credential: Name of the credential set you've just created. (i.e., Arcsight ESM Credentials).

Trust Invalid SSL Certificates: Select this if server's certificate is self-signed or not recognized by browsers.

Visible Alert Fields: You may define which alarm fields will be displayed on Incident Management Service Desk.

When you click on the Test button the following popup should be displayed if your credential and address is valid.

Click Save to complete integration.

Navigate Configuration -> Parameters and set the value of ArcSightListenerEnabled to true

Configuring Micro Focus Arcsight ESM as Integration

Navigate Configuration -> Integrations and click on Create Integration.

Fill the configuration form as follows:

Name: Display name of Micro Focus Arcsight ESM integration on SOAR.

Type: Micro Focus Arcsight ESM.

Address: Address of Micro Focus Arcsight ESM Manager (the format should be https://192.168.5.5:8443).

Configuration: You need to specify the following configuration parameters.

#proxy.id=5422

Credential: Name of the credential set you've just created. (i.e., Arcsight ESM Credentials).

Trust Invalid SSL Certificates: Select this if server's certificate is self-signed or not recognized by browsers.

Require Approval From: Select user(s) from list to ask her/his approval before executing actions on this integration.

Notify: Select user(s) from the list to notify when SOAR performs an action on this integration.

When you click on the Test button the following popup should be displayed if your credential and address is valid.

Click Save to complete integration.

Additional Notes

The following configuration parameters can be used for fine tuning the integration. Please consult SOAR field engineering team before editing them:

Parameter Name Description Default Value

ArcSightAutoEnrichEnabled Enable ArcSight auto-enrichment with base-event data false

ArcSightListenerEnabled Enable Arcsight Listener false

ArcSightListenerKeyField ArcSight listener key field for alert source identification oldFileHash

ArcSightListenerPort Arcsight listenet port 9090

ArcSightListenerProtocol ArcSight listener protocol tcp

ArcSightListenerThreadPoolCoreSize ArcSight listener thread pool core pool size (0 = unlimited)
0

ArcSightListenerThreadPoolKeepAlive ArcSight listener thread pool keep-alive seconds (ignored if core pool

size = 0)

60

ArcSightListenerThreadPoolMaxSize ArcSight listener thread pool maximum size (ignored if core pool size =

0)

20

ArcSightListenerThreadPoolQueueCapacity ArcSight listener thread pool queue capacity (ignored if core pool size =

0)

1000s

Integration Guide for Micro Focus ArcSight Intelligence

Integration Overview

Micro Focus ArcSWight Intelligence is using unsupervised machine learning to calculate probabilistic risk assessments based on behavioral analytics from millions of events, ultimately generating a short list of high value targets to allow security teams to detect, investigate, and respond to threats that may be hiding in the enterprise before any incident occurs.

Integration Capabilities

SOAR has the following integration capabilities with Micro Focus ArcSight Intelligence:

- Ingest Anomalies as Alert
- Get Entity Details

Use Case #1: Incident Prioritization

SOAR is integrated with Micro Focus ArcSight Intelligence, to help the prioritization and

investigation of incidents as well as remediation of incidents. When an alert comes a new incident is created on SOAR's own Incident Management Service Desk. SOAR then automatically checks the risk scores of entities and prioritise the incident based on these risk scores.

Use Case #2: Mitigating Account Compromise

SOAR ingests anomaly data from ArcSight Intelligence and create an incident ticket on its own Incident Management Service Desk. With its broad integration portfolio, orchestration, and automation capabilities, SOAR investigates, ascertains the case, and takes necessary actions to prevent the compromise.

Configuration

Prerequisites

- SOAR connects to Micro Focus ArcSight Intelligence API via HTTPS. By default interface works on 443/tcp port. So access permission to this port is required.
- A user account for SOAR to connect to Micro Focus ArcSight Intelligence API.

Configuring ArcSight Intelligence

No specific configuration is needed on Micro Focus ArcSight Intelligence.

Configuring SOAR

1. Click **Configuration** > **Credentials** > **Create Credential**.
2. Specify the following parameter values in the **Credential Editor**:
   a. **Internal Credential**

Parameter	Value
Type	Internal Credential
Name	Display name of credential set (For example, Micro Focus ArcSight Intelligence Credentials)
Username	User you have created for SOAR on Micro Focus ArcSight Intelligence..
Password	Password of the user you have created for SOAR on Micro Focus ArcSight Intelligence.
Private Key	Empty

   b. **Credential Store:**

Parameter	Value
Type	External Credential
Name	Name of the credential with pull path of the safe on store

Configuring Micro Focus ArcSight Intelligence as Alert Source

1. Click **Configuration** > **Integrations** > **Create Integration**.
2. Specify the following parameter values in the **Configuration** form:

Parameter	Value
Name	Display name of Micro Focus ArcSight Intelligence Alert Source on SOAR.
Type	Micro Focus ArcSight Intelligence.
Address	Address of the Micro Focus ArcSight Intelligence server (the format must be https://172.16.11.9).

Configuration	Specify the following configuration parameters:
	`# Tenant id`
	`tenant.id=`
	`# Minimum risk of anomaly, default value is 25. Value must be between 0-100 (lowest to highest)`
	`min.risk=75`
	`# Scope fields to be extracted from correlated events (field1:CATEGORY:ROLE, field2:CATEGORY:# CATEGORY is any of: EMAIL_ADDRESS, HASH, HOST, MAC_ ADDRESS, NETWORK_ADDRESS,`
	`# COMPUTER_NAME, UNKNOWN, URL, USERNAME, PROCESS`
	`# ROLE is any of: OFFENDER, IMPACT, RELATED`
	`#`
	`# Note: Field names must start with / character`
	`#`
	`# Example: correlated.scope=/sourcev6:NETWORK_ADDRESS:OFFENDER`
	`# correlated.scope=`
	`# How far (in days) into the past SOAR will look for alerts at the initial sync task`
	`# If not provided, SOAR will use 14 days by default`
	`days.to.look.back.at.initial.sync=14`
	`# ID of the proxy integration to use when connecting to current source.`
	`# If not provided, SOAR will try to use a direct connection.`
	`#proxy.id=123`
	`# To use basic authentication, use.basic.authentication=true`
	`# If not provided, SOAR will consider this propert as false`
	`# If use.basic.authentication is false, client.id and client.secret must be filled`
	`use.basic.authentication=false`
	`# Base path of the Interset. We are adding it to end of the URL to access Interset.`
	`interset.context.path=/interset`
	`# Client id that defined in OSP`
	`client.id=id`
	`# Client secret that defined in OSP`

	`client.secret=secret`
Credential	Name of the credential set you have created (For example, Micro Focus ArcSight Credentials Credentials)
Trust Invalid SSL Certificates	Select this if Web UI's certificate certificate is self-signed or is not recognized by browsers
Visible Alert Fields	You may define which alarm fields will be displayed on Incident Management Service Desk.

3. Click **Test** to test the integration.
4. Click **Save** to complete the integration.

Configuring Micro Focus ArcSight Intelligence as Integration

1. Click **Configuration** > **Integrations** > **Create Integration**.
2. Specify the following parameter values in the **Configuration** form:

Parameter	Value
Name	Display name of Micro Focus ArcSight Intelligence integration on SOAR.
Type	Micro Focus ArcSight Intelligence.
Address	Address of the Micro Focus ArcSight Intelligence server (the format must be https://172.16.11.9).

Configuration	Specify the following configuration parameters:
	`# Tenant id`
	`tenant.id=`
	`# ID of the proxy integration to use when connecting to current source.`
	`# If not provided, SOAR will try to use a direct connection.`
	`#proxy.id=123`
	`# configure how far (in minutes) into the past this enrichment will look.`
	`#cache.reusing.duration=20`
	`# To use basic authentication, use.basic.authentication=true`
	`# If not provided, SOAR will consider this propert as false`
	`# If use.basic.authentication is false, client.id and client.secret must be filled`
	`use.basic.authentication=false`
	`# Base path of the Interset. We are adding it to end of the URL to access Interset.`
	`interset.context.path=/interset`
	`# Client id that defined in OSP`
	`client.id=id`
	`# Client secret that defined in OSP`
	`client.secret=secret`

Credential	Name of the credential set you have created (For example, Micro Focus ArcSight Credentials Credentials)
Trust Invalid SSL Certificates	Select this if Web UI's certificate certificate is self-signed or is not recognized by browsers
Require Approval From	Select user(s) from list to ask her/his approval before executing actions on this integration. Since SOAR only executes enrichments on Interset, leave it empty.
Notify	Select user(s) from the list to notify when SOAR performs an action on this integration. Since SOAR only executes enrichments on ArcSight Intelligence, leave it empty.

3. Click **Test** to test the integration.
4. Click **Save** to complete the integration.

Additional Notes

- Get Entity Details enrichment results return latest 1000 records in maximum.
- The following configuration parameters can be used for fine tuning the integration. You must consult SOAR field engineering team before editing them:
  - MicroFocusIntersetListenerMaxRetrySeconds Micro Focus Interset listener queue max message retry in seconds 1800
  - MicroFocusIntersetListenerQueueConcurrency Upper limit of Micro Focus Interset Listener consumer thread count 3
  - MicroFocusIntersetSyncPeriod Period in seconds to sync Micro Focus Interset anomalies 60

Integration Guide for Micro Focus Arcsight Logger

Integration Overview

Arcsight Logger is a log management solution for compliance, efficient log search, and secure storage.

Integration Capabilities

ArcSight SOAR has the following integration capability with Micro Focus Arcsight Logger:

- Search Query

Use Case: Investigating Cyber-attacks

Integrated with Micro Focus Arcsight Logger, ATAR queires logs collected from various enterprise systems to enrich incident ticket, and improve analyst's understanding of incident.

Configuration

Prereqisites

- Currently SOAR supports Micro Focus Arcsight Logger version 6.3.1.7874.0 and later.SOAR connects to Micro Focus Arcsight Logger API using HTTPS. By default REST-API interface works on 443/tcp port. So access permission to this port is required.
- A user account is required for ATAR to connect Micro Focus Arcsight Logger.

Configuration on Micro Focus Arcsight Logger

- Click **System Admin** > **Users/Groups** > **User Management** and add a user account with **Default Logger Search Group**.

Configuring SOAR

Configuring SOAR

1. Click **Configuration** > **Credentials** > **Create Credential**.
2. Specify the following parameter values in the **Credential Editor**:

a. **Internal Credential**

Parameter	Value
Type	Internal Credential
Name	Display name of credential set (For example, Arcsight Logger Credentials)
Username	User you have created for ATAR on Micro Focus Arcsight Logger.
Password	Password of the user you have created for ATAR on Micro Focus Arcsight Logger.
Private Key	Empty

b. **Credential Store:**

Parameter	Value
Type	External Credential
Name	Name of the credential with pull path of the safe on store

3. Click **Configuration** > **Integrations** > **Create Integration**.
4. Specify the following parameter values in the **Configuration** form:

Parameter	Value
Name	Display name of Micro Focus Arcsight Logger integration on SOAR
Type	Micro Focus Arcsight Logger
Address	Address of the integration (the format must be https://192.168.12.6)

Configuration	Specify the following configuration parameters:
	`events.pageLength=10000`
	`# configure how far (in minutes) into the past this enrichment will look.`
	`#cache.reusing.duration=20`
	`# local search enabling parameter for Search Query capability.`
	`# If this is set false, ATAR will perform searches on all nodes.`
	`#local.search.enabled=false`
	`# use master session while fetching events from peers for Search Query.`
	`# If this is set true, ATAR will use the same session ID while performing`
	`# searches on the other nodes.`
	`#reuse.master.session=false`
	`# peers credential list (if master session won't be shared)`
	`# peer address and credential ID values must be separated with :`
	`# additional peer-credential pairs must be separated with |`
	`#peer.credential.list=1.1.1.1:CredentialId|2.2.2.2:CredentialId`

Credential	Name of the credential set created on step 2 (For example, Arcsight Logger Credentials)
Trust Invalid SSL Certificates	Select this if Engine's certificate is self-signed or is not recognized by browsers

5. Click **Test** to test the integration.
6. Click **Save** to complete the integration.

Additional Notes

- In order to execute queries on Micro Focus Arcsight Logger, you should create query scripts with **ArcSight Logger Query** type under **Configuration** -> **Customization Library**.
- SOAR extracts scope items on columns defined as Artifact in the query script. For example, `// Artifact: deviceCustomNumber1Label | KEYWORD | RELATED`

Integration Guide for Microsoft Active Directory

Integration Overview

Active Directory is an umbrella title for directory-based identity related services that Microsoft developed for the Windows domain networks.

ArcSight SOAR has the following integration capabilities with Microsoft Active Directory:

- Authenticate SOAR users from AD
- Add user to a group
- Remove user from a group
- Lock user acoount
- Get user information
- Get user's groups
- Get group list
- Get group information
- Get computer information
- List computers on domain
- Fetch a domain object

Use Case: Compromised user account

During the investigation of the attack SOAR can ask Microsoft Active Directory the details of theuser account suspicious to be compromised, check the groups account belongs to, locks the account, fetches her/his manager's information and send a notification e-mail to manager if needed.

This can be performed automatically within a playbook or manually by an analyst.

Configuration

Prerequisites

- SOAR connects to Microsoft Active Directory using LDAPS protocols. Access to 636/tcp
- port is required.
- A domain user account is required for SOAR to connect Microsoft Active Directory.

Configuration on Microsoft Active Directory

- Create a user account on Domain Controller with no password expiry.
- Add this user into “Account Operators” group. Members of this group can manage groups and accounts on domain except domain admins.

Configuring SOAR

1. Navigate **Configuration** > **Credentials** and click **Create Credential**.
2. Fill the Credential Editor form as follows:

   a. Internal Credential:

   Type: Internal credential.

   Name: Display name of credential set (i.e., Microsoft AD Credentials).

   Username: User you have created for SOAR on Microsoft Active Directory (the format should be username@domain).

   Password: Password of the user you have created for SOAR on Microsoft ActiveDirectory.

   Private Key: Empty.

   b. Credential Store:

   Type: External credential.

   Name: Name of the credential with pull path of the safe on store.
3. Navigate to **Configuration** > **Integrations** and click **Create Integration.**
4. Fill the configuration form as follows:

   Name: Display name of Microsoft Active Directory integration on SOAR.

   Type: Microsoft Active Directory.

   Address: Address of the integration (the format should be 192.168.2.2:636).

   Configuration: You need to specify the following configuration parameters.

```
# SOAR will search objects under LDAP searchbase specified.
# Format should be "DC=EXAMPLE,DC=COM"
ldap.searchbase=DC=EXAMPLE,DC=COM
# LDAP domain should be like "example.com"
ldap.domain=example.com
# LDAP NT domain name should be like "EXAMPLE"
ldap.ntdomain=EXAMPLE
```

```
# Username for LDAP service availability check.
# SOAR will try to bind LDAP service as this user.
ldap.checkavailabilityuser=testuser01@example.com
# SOAR Role for newly created LDAP users. Default role is "Empty Role"
ldap.new.user.default.role.id="Empty Role"
# LDAP property name of mobile number. SOAR will use this value to send
# notifications if needed. Default value is "mobile".
ldap.property.mobile=mobile
# Set this parameter true if this integration will be used to
authenticate
# SOAR users
login.realm=true
# Define LDAP group of users who can authenticate to SOAR
login.ldap.group=ATARUsers
# configure how far (in minutes) into the past this enrichment will look.
cache.reusing.duration=30
```

Credential: Name of the credential set you've just created on step 2. (i.e., Microsoft AD Credentials).

Trust Invalid SSL Certificates: Select this if Engine's certificate is self-signed or not recognized by browsers.

Require Approval From: Select user(s) from list to ask her/his approval before executing actions on this integration.

Notify: Select user(s) from the list to notify when SOAR performs an action on this integration.

5. Click on the **Test** button.
6. Click **Save** to complete integration.

Additional Notes

In order to use configured Microsoft Active Directory integration to authenticate SOAR users:

Navigate to **Configuration** > **Parameters** and edit RESTAPILDAPAuthenticationIntegrationID parameter. Set its value to ID of the Microsoft Active Directory integration you have already defined on previous step.

Integration Guide for Microsoft Exchange

Integration Overview

Exchange Server is a mail server developed by Microsoft.

SOAR has the following integration capabilities with Microsoft Exchange Server :

- Delete email
- Mark email
- Quarantine email

Use Case: Deleting already delivered phishing emails

SOAR can follow email inboxes for user's phishing reports and automatically creates an incident record on its service desk. During the investigation of the attack SOAR can extract the sender address and subject and using these values performs a search on Microsoft Exchange Server to mark or delete already delivered malicious messages. This can be performed automatically within a playbook or manually by an analyst.

Configuration

Prerequisites

- SOAR connects to Microsoft Exchange Web Service API via HTTPS. So access to 443/tcp port is required.
- A user account with impersonation role is required for SOAR to connect Microsoft Exchange.

Configuration on Microsoft Exchange

1. Login to Microsoft Exchange admin center and add a user mailbox for SOAR.
2. Open Exchange Management Shell and give the user Application Impersonation role using the following command:
3. New-ManagementRoleAssignment \
4. -Name:<impersonation Assignment Name> \
5. -Role:ApplicationImpersonation \
6. -User:<account name>
7. Configuring SOAR

8. Navigate Configuration -> Credentials and click on Create Credential.
9. Fill the Credential Editor form as follows:
10. a. Internal Credential:
11. Type: Internal credential.
12. Name: Display name of credential set (i.e., Microsoft Exchange Credentials).
13. Username: User you have configured SOAR on Microsoft Exchange (the format should
14. be username@domain).
15. Password: Password of the user you have configured for SOAR on Microsoft
16. Exchange.
17. Private Key: Empty.
18. b. Credential Store:
19. Type: External credential.
20. Name: Name of the credential with pull path of the safe on store.
21. Navigate Configuration -> Integrations and click on Create Integration.
22. Fill the configuration form as follows:
23. Name: Display name of Microsoft Exchange integration on SOAR.
24. Type: Microsoft Exchange.
25. Address: Address of the integration (the format should be 192.168.2.8).
26. Configuration : You need to specify the following configuration parameters.
27. requests.impersonation.disable=false
28. requests.cookies.enable=true
29. mail.store.protocol=exchange
30. mail.incoming.pollerperiod=10000
31. mail.incoming.folder=Inbox
32. Credential: Name of the credential set you've just created on step 2. (i.e., Microsoft
33. Exchange Credentials).
34. Trust Invalid SSL Certificates: Select this if certificate used on Exchange Server
35. is self-signed or not recognized by browsers.
36. Require Approval From: Select user(s) from list to ask her/his approval before
37. executing actions on this integration.
38. Notify: Select user(s) from the list to notify when SOAR performs an action on this
39. integration.
40. When you click on the Test button the following popup should be displayed if your

41. credential and address is valid.
42. Click Save to complete integration.

Additional Notes

- To customize warning messages for Quarantine and Mark actions, edit the following parameters under **Configuration** > **Parameters**:
  - MSExchangeMarkWarningText
  - MSExchangeQuarantineWarningText
- To customize the mail folder to be used for Quarantine actions, edit the following parameter under **Configuration** > **Parameters**:
  - MSExchangeQuarantineEMailBox
- In some environments with multiple CAS deployments Exchange uses a request cookie totrack the environment. requests.cookies.enable configuration should help track the cookie so that SOAR won't have any mismatch and Subscription was not found error. It is by default true and should stay that way in most environments.

Integration Guide for Microsoft Office365 Exchange EWS

Integration Overview

Exchange Server EWS provide access to mailbox data stored in Exchange Online, Exchange Online as part of Office 365, and on-premises versions of Exchange starting with Exchange Server 2007, and enable you to manage that information according to the requirements of your organization.

Note: This is the new version of Microsoft Exchange integration and old one will be phased out.

Users are encouraged to use this integration.

ArcSight SOAR has the following integration capabilities with Microsoft Exchange EWS :

- Block Email Sender
- Delete Email
- Delete Attachment
- Get Attachments
- Get Emails
- Search Emails

Use Case: Deleting already delivered phishing emails

SOAR follows email inboxes for user's phishing reports and automatically creates an incident record on its service desk. During the investigation of the attack ATAR can extract the sender address and subject and using these values performs a search on Microsoft Exchange Server to delete already delivered malicious messages and block malicious senders. This can be performed automatically within a playbook or manually by an analyst.

Configuration

Prerequisites

- SOAR connects to Microsoft Exchange Web Service API using HTTPS. So access to 443/tcp port is required.

- A user account with the following permissions is required for SOAR to connect MS Exchange EWS Server:
  - ApplicationImpersonation (Authorized to make operations for other users' accounts)
  - MailboxSearch (Authorized to search all mailboxes).

Configuration on Microsoft Exchange

1. Login to Microsoft Exchange Admin Center (For example, https://exchangeserver/ecp) and add a user mailbox for SOAR.
2. Navigate to **Permissions** > **Cloud Migrator Impersonation**, edit and add user account you have created in first step to "Members" to give Account Impersonation permission.
3. Navigate to **Permissions** > **Discovery Management**, edit and add user account you have created in first step to "Members" to give Mailbox Search permission.

Configuring SOAR

1. Navigate **Configuration** > **Credentials** and click **Create Credential**.
2. Fill the Credential Editor form as follows:

   a. Internal Credential:

   Type: Internal credential.

   Name: Display name of credential set (i.e., MS Exchange EWS Credentials).

   Username: User you have configured ATAR on Microsoft Exchange (the format should be username@domain).

   Password: Password of the user you have configured for ATAR on Microsoft Exchange.

   Private Key: Empty.

   b. Credential Store:

   Type: External credential.

   Name: Name of the credential with pull path of the safe on store.
3. Navigate **Configuration** > **Integrations** and click **Create Integration**.
4. Fill the configuration form as follows:

   Name: Display name of Microsoft Exchange EWS integration on ATAR.

   Type: Microsoft Exchange EWS.

   Address: Address of the integration (the format should be outlook.office365.com or 192.168.2.7).

   Configuration : You need to specify the following configuration parameters.

   ```
   # Maximum record number per paginated response. Default value is 1000
   ```

```
page.size=200

# Connect time out in seconds. Default value is 200

connect.timeout=7200

# Request time out in seconds. Default value is 200

request.timeout=7200

# Trash folder name. Default value is Deleted Items

#trash.folder=

# Junk folder name. Default value is Junk Email

#junk.folder=

# Maximum record number per paginated attachment detail response. Default
value is 10

#attachment.page.size=

# Microsoft Exchange Server enrichment API timezone, if not specified GMT
will be used as default

#timezone=

# Maximum number of email id list per request. Default value is 5

#email.id.size=

# Maximum record number per paginated item detail response. Default value
is 10

#email.page.size=

# Maximum email item limit for each enrichment. Default value is 1000

#email.limit=

# Maximum attachment item limit for each enrichment. Default value is 100

#attachment.limit=
```

Credential: Name of the credential set you've just created on step 2. (i.e., Microsoft Exchange Credentials).

Trust Invalid SSL Certificates: Select this if certificate used on Exchange Server is self-signed or not recognized by browsers.

Require Approval From: Select user(s) from list to ask approval before executing actions on this integration.

Notify: Select user(s) from the list to notify when ATAR performs an action on this integration.

5. Click the Test button.
6. Click **Save** to complete integration.

Additional Notes

For Delete capability, at least one of the following parameters should be given:

- Email From
- Email Subject
- Email ID
- Attachment ID

And there are 3 deletion methods:

- **Hard Delete:** Deletes permanently (default)
- **Move To Trash:** Moves to trash folder (such as Deleted Items folder)
- **Soft Delete:** Moves to dumpster if it is enabled.

Integration Guide for Microsoft Windows DNS Server

Integration Overview

ArcSight SOAR uses Microsoft Windows DNS Server to redirect IP address to another IP address.

SOAR checks connection.secure parameter to connect via WinRM over http or https protocol.

Integration Capabilities

- Action
- Block

Configuration

Configuration on Microsoft Windows DNS Server

- SOAR connects to Microsoft Windows DNS Server's integration API via WinRM services. Therefore SOAR should be able to connect this service.
- WinRM credential is required.

Configuring ATAR

1. While creating this integration via Integrations tab of Configuration menu:

   Name: Display name of the integration.

   Type: Microsoft Windows DNS Server.

   Address: Address of the integration (the format should be http[s]://1.1.1.1:1234).

   Credential: WinRM credential is required. Credential that has been defined for this integration under the **Credentials** menu.

   Configuration: You need to specify the following configuration parameters.

   ```
   dns.zone.name: Redirected DNS server zone name
   ```

   ```
   dns.block.ip: Redirection address
   ```

```
dns.server.name: DNS server name

#Use https:// instead of http:// on WinRM connection

connection.secure=true : For secure connections, otherwise set to false.

#Parameters:
```

WindowsDNSCommandExecPath: Windows DNS command execution path.

Trust Invalid SSL Certificates: Select this if Engine's certificate used for the service is self-signed or not recognized by browsers.

Require Approval From: Select user(s) from list to ask her/his approval before executing actions on this integration.

Notify: Select user(s) from the list to notify when ATAR performs an action on this integration.

2. Click the **Test** button.
3. Click **Save** to complete integration.

Integration Guide for Microsoft Windows Services (WinRM)

Integration Overview

Integration Capabilities

- Action
- None

Configuration

Configuration on Microsoft Windows Services

- SOAR connects to Microsoft Windows Service's integration API via WinRM services.
- Therefore SOAR should be able to connect this service.
- WinRM credential is required.

Configuring SOAR

1. While creating this integration via Integrations tab of Configuration menu:

   Name: Display name of the integration.

   Type: Microsft Windows Services.

   Address: Address of the integration (the format should be 1.1.1.1 or abc.example.com).

   Configuration: You need to specify the following configuration parameters.

   putfile.generateuuid =

   putfile.defaultfolder =

   connection.secure = true

   Credential: Credential that has been defined for this integration under the Credentials menu.

   Trust Invalid SSL Certificates: Select this if certificate used for the service is selfsigned or not recognized by browsers.

Require Approval From: Select user(s) from list to ask her/his approval before executing actions on this integration.

Notify: Select user(s) from the list to notify when SOAR performs an action on this integration.

2. Click the **Test** button.
3. Click **Save** to complete integration.

Integration Guide for MISP

Integration Overview

The MISP threat sharing platform is a free and open source software helping information sharing of threat intelligence including cyber security indicators.

Integration Capabilities

ArcSight SOAR has the following integration capabilities with MISP.

- File Reputation
- IP Reputation
- URL Reputation
- Get Event
- Add Attribute to Event
- Add Tag to Event
- Create Event
- Create Event with Attribute
- Remove Attribute from Event
- Remove Tag from Event

ArcSight SOAR integrates with MISP to gather, store threat information and can query to IoCs. The capabilities can either be performed automatically within a playbook or manually by an analyst.

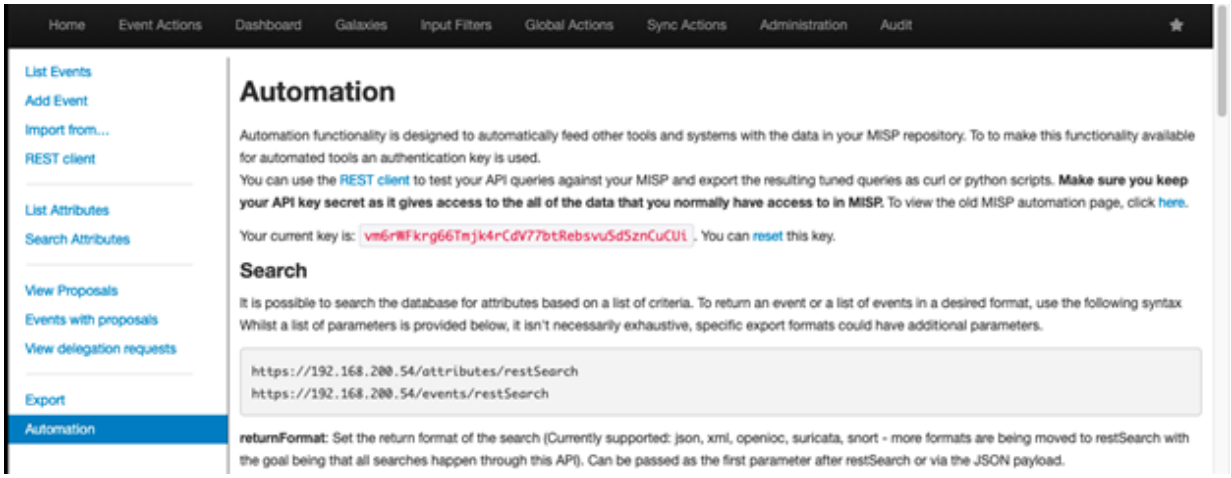
Configuration

Prerequisites

• Access to tcp port 443 as SOAR connects to MISP using HTTPS

• An API key for SOAR to connect to MISP

Note: To gather the API key for SOAR, navigate to **MISP Interface** > **Event Actions** > **Automation**.



Configuring SOAR

1. Click **Configuration** > **Credentials** > **Create Credential**.
2. Specify the following parameter values in the **Credential Editor** form:
   a. **Internal Credential**

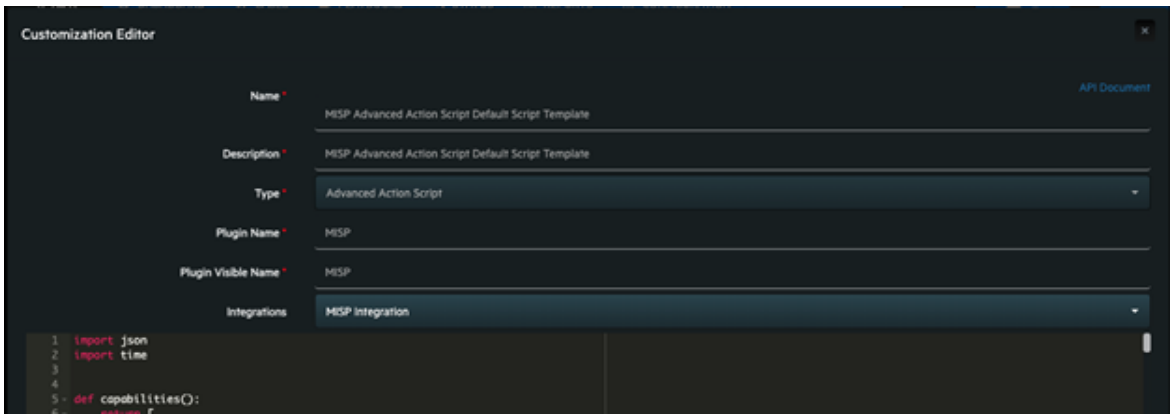
Parameter	Value
Type	Internal Credential
Name	Display name of credential set (For example, MISP Credentials)
Username	Empty
Password	Empty
Private Key	API Key retrieved from the MISP

3. Click **Configuration** > **Integrations** > **Create Integration**.
4. Specify the following parameter values in the **Configuration form**:

Parameter	Value
Name	Display name of MISP integration on ArcSight SOAR
Type	MISP
Address	Address of the cloud service, in the following format: https://<misp_ environement_ip>
Credential	Name of the credential set created in the previous step(For example, MISP Credentials)

Trust Invalid SSL Certificates	Not Applicable
Require Approval From	Select users from the list who can provide approval before executing enrichments on the integration
Notify	Select users from the list to notify when SOAR performs an enrichment on the integration

5. Click **Save** to complete the integration.
6. Navigate to **Configuration** > **Customization Library**.
7. In the **Customization Editor**, Edit **MISP Advanced Action Script Default Script Template** and for the **Integrations** field select the integration you saved (for example, MISP Integration).



8. Navigate to **Configuration** > **Integrations**.
9. Click **Edit** for the MISP integration you created.
10. Click **Test** to test the integration.

Integration Guide for Ones BioAffix

Integration Overview

Ones BioAffix is a biometric single sign on (Biometric SSO) and biometric identity verification solution which lets organizations to manage their physical security and access.

Integration Capabilities

ArcSight SOAR has the following integration capability with Ones BioAffix:

- Change User Status (Block & Unblock)
- User Details (Info & Logs)

Use Case: Blocking Suspicious Employees

Integrated with Ones BioAffix ATAR lets users to investigate suspicious employee traffic through building and block access if needed. This can be performed automatically within a playbook or manually by an analyst.

Configuration

Prerequisites

- Currently SOAR supports Ones BioAffix version 4.20.10.1.
- SOAR connects to Ones BioAffix API via HTTPS. Typically it runs on 8443/tcp* port. So access to this service is required.
- Credentials of administrator is required for SOAR to connect Ones BioAffix.

Configuration on Ones BioAffix

- No specific configuration is needed on Ones BioAffix server.

Configuring SOAR

1. Navigate **Configuration** > **Credentials** and click **Create Credential**.
2. Fill the Credential Editor form as follows:

   a. Internal Credential:

Type: Internal credential.

Name: Display name of credential set (i.e., Ones BioAffix Credentials)

Username: Administrator username you have on Ones BioAffix.

Password: Password for the administrator user you have on Ones BioAffix.

Private Key: Empty.

b. Credential Store:

Type: External credential.

Name: Name of the credential with pull path of the safe on store.

3. Navigate to **Configuration** > **Integrations** and click **Create Integration**.
4. Fill the configuration form as follows:

   Name: Display name of Ones BioAffix integration on ATAR.

   Type: Ones BioAffix Server.

   Address: Address of the integration (the format should be https://192.168.12.77:8443).

   Credential: Name of the credential set you've just created on step 2. (i.e., Ones BioAffix Credentials).

   Trust Invalid SSL Certificates: Select this if Engine's certificate is self-signed or not recognized by browsers.

   Require Approval From: Select user(s) from list to ask her/his approval before executing actions on this integration.

   Notify: Select user(s) from the list to notify when ATAR performs an action on this integration.
5. Click the **Test** button.
6. Click **Save** to complete integration.

Additional Notes

Due to API behaviour of Ones BioAffix integration, "Date of Birth", "Phone" and "Profile Photo" of users should be set to execute actions.

Integration Guide for Palo Alto Networks AutoFocus

Integration Overview

Palo Alto Networks AutoFocus is a threat intelligence platform which allows to search attack indicators and access to details of them. AutoFocus provides the intelligence, analytics, and context required to understand which attacks require immediate response and take decisive action to prevent future attacks.

Integration Capabilities

ArcSight SOAR has the following integration capabilities with Palo Alto Networks AutoFocus:

- Search Email Address
- Search File Hash
- Search File Name
- Search IP Address
- Search URL

Use Case: Investigating Phishing Campaigns

SOAR integrates with Palo Alto Networks AutoFocus to search attack indicators. SOAR can follow email inboxes for user's phishing reports and automatically creates an incident record on its service desk. During the investigation of the attack SOAR can extract the sender address, IP address, files in the attachment and ask these indicators to Palo Alto Networks AutoFocus if this is a known attack and previously analyzed. This can be performed automatically within a playbook or manually by an analyst.

Configuration

Prerequisites

- SOAR connects to Palo Alto Networks AutoFocus API via HTTPS. Access (https://autofocus.paloaltonetworks.com (443/tcp port) is required.
- An API key is required for SOAR to connect Palo Alto Networks AutoFocus.

Configuration on Palo Alto Networks AutoFocus

No specific configuration is needed. Login to https://autofocus.paloaltonetworks.com and note the API key under **Settings** > **General** menu.

Configuring SOAR

1. Navigate **Configuration** > **Credentials** and click **Create Credential**.
2. Fill the Credential Editor form as follows:

   a. Internal Credential:

   Type: Internal credential.

   Name: Display name of credential set (i.e., PAN AutoFocus Credential).

   Username: Empty.

   Password: API Key.

   Private Key: Empty.

   b. Credential Store:

   Type: External Credential.

   Name: Name of the credential with pull path of the safe on store.
3. Navigate **Configuration** > **Integrations** and click **Create Integration**.
4. Fill the configuration form as follows:

   Name: Display name of Palo Alto Networks AutoFocus integration on SOAR.

   Type: Palo Alto Networks AutoFocus.

   Address: Address of the integration (https://autofocus.paloaltonetworks.com).

   Credential: Name of the credential set you've just created on step 2. (i.e., PAN AutoFocus Credential).

   Configuration: You need to specify the following configuration parameters

```
# Integration ID of the proxy integration to use when connecting to
# current integration.
# If not provided, SOAR will try to use a direct connection.
#proxy.id=123
# configure how far (in minutes) into the past this enrichment will look.
# cache.reusing.duration=20
```

Trust Invalid SSL Certificates: Select this if Engine's certificate is self-signed or not recognized by browsers.

Require Approval From: Select user(s) from list to ask approval before executing actions on this integration.

Notify: Select user(s) from the list to notify when SOAR performs an action on this integration.

5. The **EnrichmentFixedDelay** configuration parameter value must be set to less than 120 seconds because of AutoFocus' requirement. Otherwise AutoFocus API cookie will be expired.
6. Click the **Test** button.
7. Click **Save** to complete integration.

Integration Guide for Palo Alto Networks Firewall

Integration Overview

Palo Alto Networks Next Generation Firewall is a security technology that combines firewall, antivirus, intrusion prevention, and virtual private network (VPN) capabilities to provide proactive threat defense that stops attacks before they spread through the network.

Integration Capabilities

ArcSight SOAR has the following integration capabilities with Palo Alto Firewall (API):

- Block IP
- Block Host
- Block URL
- Disconnect

Use Case: Blocking access to malicious IP addresses and hosts

Integrated with Palo Alto Networks NGFW, SOAR blocks malicious IP addresses and hosts on perimeter while responding cyber-attacks. Blocking can be performed automatically within a playbook or manually by an analyst.

Configuration

Prerequisites

- Currently SOAR supports Palo Alto Networks NGFW 9.0.1 version.
- SOAR connects to Palo Alto Networks NGFW API via HTTPS. Access to 443/tcp port is required.
- An API key is required for SOAR to connect Palo Alto Networks Firewall.

Configuration on Palo Alto Networks Firewall (API)

1. Navigate Device menu and create a new Admin Role for SOAR. New role must be restricted to only specific XML API operations. Only required permissions are: "Configuration", "Operational Requests" and "Commit".

2. Do not forget to disable all Web UI and Command Line permissions since they are unnecessary.
3. Create an Administrator account with SOAR API Role you have created in first step.
4. Navigate to Objects > Address Groups and add an address group for IPs to be populated by SOAR actions.
5. Similarly add an address group for hosts/FQDNs to be populated by SOAR.
6. Navigate ****Objects** > **Custom** Objects and add a Custome URL Category to be populated by SOAR.
7. Commit all changes.
8. To obtain API key run the following request from command line.

```
curl -k -X GET 'https://PaloAlto_NGFW_IP/api/?type=keygen& \
user=atarapi&password=password'
```

Configuring SOAR

1. Navigate **Configuration** > **Credentials** and click **Create Credential**.
2. Fill the Credential Editor form as follows:

   a. Internal Credential:

   Type: Internal credential.

   Name: Display name of credential set (i.e., Palo Alto Firewall Credential).

   Username: User you have created for SOAR on Palo Alto NGFW.

   Password: Password of the user you have created for SOAR on Palo Alto NGFW.

   Private Key: API Key you have created for SOAR.

   b. Credential Store:

   Type: External credential.

   Name: Name of the credential with pull path of the safe on store.
3. Navigate **Configuration** > **Integrations** and click **Create Integration**.
4. Fill the configuration form as follows:

   Name: Display name of Palo Alto Networks Firewall integration on SOAR.

   Type: Palo Alto Networks Firewall (API)

   Address: Address of the integration (the format should be https://192.168.2.78).

   Credential: Name of the credential set you've just created on step 2. (i.e., Palo Alto Firewall Credential).

Trust Invalid SSL Certificates: Select this if web UI's certificate is self-signed or not recognized by browsers.

Configuration: You need to specify the following configuration parameters.

```
# Address group to use when blocking IP addresses.

# This address group should be created in Palo Alto device before use.

addressgroup.ip=ATAR_BLOCK_IP

# Address group to use when blocking host names.

# This address group should be created in Palo Alto device before use.

addressgroup.host=ATAR_BLOCK_HOST

# Custom URL category to use when blocking URLs.

# This custom URL category should be created in Palo Alto device before
use.

custom.url.category=ATAR_BLOCK_URL
```

Require Approval From: Select user(s) from list to ask her/his approval before executing actions on this integration.

Notify: Select user(s) from the list to notify when SOAR performs an action on this integration.

5. Click on the Test button.
6. Click **Save** to complete integration.

Additional Notes

Palo Alto Networks NGFW integration supports multiple "vsys". If your firewall has more than one "vsys" SOAR will ask you to choose one while taking action.

Integration Guide for Palo Alto Networks Panorama

Integration Overview

The Panorama management server provides centralized monitoring and management of multiple Palo Alto Networks next-generation firewalls and of WildFire appliances and appliance clusters.

Integration Capabilities

ArcSight SOAR has the following integration capabilities with Palo Alto Networks Panorama:

- Block IP address
- Block Host
- Block URL

Use Case: Blocking malicious IP addresses on multiple firewall appliances

With this integration, SOAR can block malicious IP addresses, hosts and URL addresses on multiple firewall devices simultaneously while responding cyber-attacks. This can be performed automatically within a playbook or manually by an analyst.

Configuration

Prerequisites

- Currently SOAR supports Palo Alto Network Panorama 8.1.0 version.
- SOAR connects to Palo Alto Networks Panorama API using HTTPS. Access to 443/tcp port is required.
- An API key is required for SOAR to connect Palo Alto Networks Panorama.
- If users want to use multiple devicegroup, they should write devicegroup names comma separated, for ex: Ankara, Istanbul, Izmir

Configuration on Palo Alto Networks Panorama

1. Navigate to **Panorama** menu and create a new Admin Role for SOAR. The new role should be

   restricted to only specific XML API operations. Only required permissions are: "Configuration", "Operational Requests" and "Commit". Do not forget to disable all Web UI and Command Line permissions since they are unnecessary.
2. Create an Administrator account with **Custom Panorama Admin** type and SOAR API Role you have created in first step.
3. Commit all changes.
4. In order to obtain API key run the following request from command line.

```
curl -k -X GET 'https://Panorama_IP/api/?type=keygen& \
```

```
user=atarapi&password=password'
```

Configuration on SOAR

1. Navigate **Configuration** > **Credentials** and click **Create Credential**.
2. Fill the Credential Editor form as follows:

   a. Internal Credential:

   Type: Internal credential.

   Name: Display name of credential set (i.e., PAN Panorama Credential).

   Username: Empty.

   Password: Empty.

   Private Key: API Key you have created for SOAR.

   b. Credential Store:

   Type: External credential.

   Name: Name of the credential with pull path of the safe on store.
3. Navigate **Configuration** > **Integrations** and click **Create Integration**.
4. Fill the configuration form as follows:

   Name: Display name of Palo Alto Networks Panorama integration on SOAR.

   Type: Palo Alto Networks Panorama.

   Address: Address of the integration (https://10.0.2.254).

   Credential: Name of the credential set you've just created on step 2. (i.e., PAN Panorama Credential).

Trust Invalid SSL Certificates: Select this if Engine's certificate is self-signed or not recognized by browsers.

Configuration: You need to specify the following configuration parameters.

```
# Device group to use when adding and address object.

# This device group should be created in Palo Alto device before use.

# If users want to use multiple devicegroups, they should write
devicegroup

# names comma separated, for ex: Ankara, Istanbul, Izmir

devicegroup.name=HeadQuarters

# Address group to use when blocking IP addresses.

# This address group should be created in Palo Alto device before use.

addressgroup.ip=ATAR_BLOCK_IP

# Address group to use when blocking host names.

# This address group should be created in Palo Alto device before use.

addressgroup.host=ATAR_BLOCK_HOST

# Custom URL category to use when blocking URLs.

# This custom URL category should be created in Palo Alto device before
use.

custom.url.category=ATAR_BLOCK_URL
```

Require Approval From: Select user(s) from list to ask her/his approval before executing actions on this integration.

Notify: Select user(s) from the list to notify when SOAR performs an action on thisintegration.

5. Click the **Test** button.
6. Click **Save** to complete integration.

Integration Guide for Recorded Future

Integration Overview

Recorded Future is a threat intelligence service which collects and analyzes vast amounts of data to deliver relevant cyber threat insights in real time.

Integration Capabilities

ArcSight SOAR has the following integration capabilities with Recorded Future:

- Lookup Domain
- Lookup Hash
- Lookup IP Address
- Lookup URL
- Lookup Vulnerability
- Search Entity Lists
- Search Malware

Use Case: Investigating Phishing Campaigns

SOAR is integrated with Recorded Future, to help investigation and mitigation of phishing campaigns. When a phishing report email comes from user, SOAR extracts the indicators such as IP address, URLs and attachments in message and a new incident is created on SOAR's own Incident Management Service Desk. SOAR then asks these indicators to Recorded Future if this is a known attack and previously analyzed. This can be performed automatically within a playbook or manually by an analyst.

Configuration

Prerequisites

- SOAR connects to Recorded Future API via HTTPS. Access to https://api.recordedfuture.com/v2/ (443/tcp port) is required.
- An API key is required for SOAR to connect Recorded Future service.

Configuration on Recorded Future

Login to https://api.recordedfuture.com/v2/ and create a new API key under user Settings > API Access menu and note the API Key and API Password generated. This token is required by SOAR to access the platform for queries.

Configuring SOAR

1. Navigate **Configuration** > **Credentials** and click **Create Credential.**
2. Fill the Credential Editor form as follows:

   a. Internal Credential:

   Type: Internal credential.

   Name: Display name of credential set (i.e., Recorded Future Credentials).

   Username: API Key you have created on Recorded Future.

   Password: API Password for the key you have created on Recorded Future.

   Private Key: Empty.

   b. Credential Store:

   Type: External credential.

   Name: Name of the credential with full path of the safe on store.
3. **Navigate Configuration > Integrations and click Create Integration.**
4. Fill the configuration form as follows:

   Name: Display name of Recorded Future integration on SOAR.

   Type: Recorded Future.

   Address: Address of the integration (https://api.recordedfuture.com/v2/).

   Configuration: You need to specify the following configuration parameters.

```
# Integration ID of the proxy integration to use when connecting to
# current integration.
# If not provided, SOAR will try to use a direct connection.
#proxy.id=123
# configure how far (in minutes) into the past this enrichment will look.
#cache.reusing.duration=20
```

   Credential: Name of the credential set you've just created on step 2. (i.e., Recorded Future Credentials)

Trust Invalid SSL Certificates: No need to select.

Require Approval From: Select user(s) from list to ask her/his approval before executing actions on this integration. Since SOAR only executes enrichments on Recorded Future, leave it empty.

Notify: Select user(s) from the list to notify when SOAR performs an action on this integration. Since SOAR only executes enrichments on Recorded Future, leave itempty.

5. Click on the **Test** button.
6. Click **Save** to complete integration.

Integration Guide for Robtex Lookup

1. Integration Overview

Robtex is used for various kinds of research of IP numbers, domain names, etc.

Robtex uses various sources to gather public information about IP numbers, domain names, host names, Autonomous systems,routes, etc. It indexes the data in a big database and provide free access for the data

2. Integration Capabilities

Action

Lookup

Configuration

Configuration on Robtex Lookup

SOAR connects to Robtex Lookup integrations via HTTPS. Therefore ATAR should be able to connect this service.

Configuring SOAR

1. While creating this integration via Integrations tab of Configuration menu:
   Name: Display name of Robtex lookup integration on SOAR.
   Type: Robtex lookup.
   Address: Address of the integration (the address should be https://www.robtex.com).
   Configuration: You need to specify the following configuration parameters

```
# Integration ID of the proxy integration to use when connecting to
# current integration.
# If not provided, ATAR will try to use a direct connection.
#proxy.id=123
```

```
# configure how far (in minutes) into the past this enrichment will look.

#cache.reusing.duration=20
```

Credential: Name of the credential set.

Trust Invalid SSL Certificates: Select this if Engine's certificate is self-signed or not recognized by browsers.

Require Approval From: Select user(s) from list to ask her/his approval before executing actions on this integration.

Notify: Select user(s) from the list to notify when ATAR performs an action on this integration.

2. Click the **Test** button.
3. Click **Save** to complete integration.

Integration Guide for Roksit DNS Firewall

Integration Overview

Roksit DNS Firewall is cloud-based cybersecurity service which provides web security and application control by analyzing DNS traffic.

Integration Capabilities

ArcSight SOAR has the following integration capability with Roksit DNS Firewall:

- Block hostname

Use Case: Blocking malicious hosts on DNS

With this integration, SOAR can block malicious hostnames on Roksit DNS Firewall service while responding cyber-attacks. This can be performed automatically within a playbook or manually by an analyst.

Configuration

Prerequisites

- SOAR connects to Roksit DNS Firewall API via HTTPS. So access to https://api.roksit.com (443/tcp port) is required.
- An API key is required to be created for SOAR to connect to Roksit DNS Firewall. Please contact to service provider.

Configuration on Roksit DNS Firewall

- No further configuration is needed.

Configuring SOAR

1. Navigate to **Configuration** > **Credentials** and click **Create Credential.**
2. Fill the Credential Editor form as follows:

   a. Internal Credential:

   Type: Internal credential.

Name: Display name of credential set (i.e., Roksit DNS FW Credentials).

Username: Empty.

Password: API Key you have obtained from Roksit.

Private Key: Empty.

b. Credential Store:

Type: External credential.

Name: Name of the credential with pull path of the safe on store.

3. **Navigate to Configuration > Integrations and click Create Integration.**
4. Fill the configuration form as follows:

   Name: Display name of Roksit DNS Firewall integration on SOAR

   Type: Roksit DNS Firewall

   Address: Address of the integration (address should be https://api.roksit.com).

   Credential: Name of the credential set you've just created on step 2. (i.e., Roksit DNS FW Credentials)

   Trust Invalid SSL Certificates: Select this if Engine's certificate is self-signed or not recognized by browsers.

   Require Approval From: Select user(s) from list to ask her/his approval before executing actions on this integration.

   Notify: Select user(s) from the list to notify when SOAR performs an action on this integration.
5. Click on the **Test** button.
6. Click **Save** to complete integration.

Additional Notes

- Roksit DNS Firewall integration on SOAR is defined as Advanced Action Script and content of the default script is accessible under **Configuration** > **Customization Library**.
- While defining the integration first time, you get a warning message as follows. For this type of integration this is the expected behaviour.

Integration Guide for RSA Envision Configuration

1. While creating this alert source via **Alert Source** tab of Configuration menu:

   Name: Display name of the alert source.

   Type: RSA Envision

   Key: Shared key between RSA Envision and SOAR.

   Allowed IP addresses: Addresses of the allowed IPs.
2. Click the **Test** button.
3. Click **Save** to complete alert source configuration.

Integration Guide for RSA Security Analytics

Integration Overview

RSA Security Analytics provides real-time visibility into network traffic with full packet capture—on premises, in the cloud and across virtual infrastructure. It helps to detect threats as they traverse in the network, monitor the timing and movement of attackers across the network and reconstruct entire network sessions to support forensic investigations.

Integration Capabilities

ArcSight SOAR has the following integration capabilities with RSA Security Analytics:

- Network Packet Capture (Time range)
- Network Packet Capture (Relative time)

Use Case: Investigating suspicious cases using packet captures

SOAR integrates with RSA Security Analytics to collect full packet capture for a given timeframe.During the investigation of an incident, SOAR can gather packet-capture from RSA Security Analytics with specified parameters such as offender IP, affected usernames, suspicious end-user machines, etc and put the related pcap file into incident timeline for further analysis and keeping evidence purposes. Collecting pcap files can be performed automatically within a playbook or manually by an analyst.

Configuration

Prerequisites

- Currently SOAR supports RSA Security Analytics version 11.0.0.0.
- SOAR connects to RSA Security Analytics Network Concentrator's API via HTTP/HTTPS.
- By default API interface works on 50105/tcp port. So access permission to this port is required.
- A user account is required to be created for SOAR to connect to RSA Security Analytics Network Concentrator API.

Configuration on RSA Security Analytics Suite

1. Login to Security Analytics Suite and navigate to **ADMIN** > **Services** and then select **Concentrator** service and open up **Security** View by clicking **Actions** icon.
2. Add a new Role to be used for SOAR user. New role should have at least “sdk.content”,“sdk.manage” and “sdk.meta” permissions".
3. Add a new user with the role you have created in previous step.

Configuring SOAR

1. Navigate **Configuration** > **Credentials** and click **Create Credential**.
2. Fill the Credential Editor form as follows:

   a. Internal Credential:

   Type: Internal credential.

   Name: Display name of credential set (i.e., RSA Security Analytics Credential).

   Username: Username you have created for SOAR on RSA Security Analytics Suite.

   Password: Password of the user you have created for SOAR on RSA Security Analytics Suite.

   Private Key: Empty.

   b. Credential Store:

   Type: External credential.

   Name: Name of the credential with pull path of the safe on store.
3. Navigate **Configuration** > **Integrations** and click **Create Integration**.
4. Fill the configuration form as follows:

   Name: Display name of RSA Security Analytics integration on SOAR.

   Type: RSA Security.

   Address: Address of the integration (the format should be http[s]://192.168.1.10:50105 or http[s]://abc.example.com:50105).

   Credential: Name of the credential set you’ve just created on step 2. (i.e., RSA Security Analytics Credential)

   Trust Invalid SSL Certificates: Select this if device’s certificate is self-signed or not recognized by browsers.

   Require Approval From: Select user(s) from list to ask her/his approval before executing actions on this integration.

Notify: Select user(s) from the list to notify when SOAR performs an action on thisintegration.

5. Click the **Test** button.
6. Click **Save** to complete integration.

Integration Guide for SMTP Mail Server

Integration Overview

ArcSight SOAR uses the SMTP Server to send emails and notification messages. ATAR can also use the same integration to access inboxes to read emails, such as device action approvals if it is configured as an IMAP server.

Integration Capabilities

- Action
- Send email

Configuration

Prerequisites

- SOAR connects to SMTP Mail Server integration via Simple Mail Transfer Protocol.Therefore SOAR must be able to connect this service.
- A user's credential is required for SMTP AUTH. The same credential will be used if IMAP is configured.

Configuring SOAR

1. Click **Configuration** > **Integrations** > **Create Integration**.
2. Specify the following parameter values in the **Configuration** form:

Parameter	Value
Name	Display name of the SMTP Mail Server integration..
Type	SMTP Mail Server
Address	Address of the integration (the format should be 1.1.1.1 or abc.example.com).

Configuration	Specify the following configuration parameters:
	`mail.default-encoding is the encoding format of emails.`
	`mail.transport.protocol is the default message transport protocol.`
	`mail.smtp.auth specifies whether SMTP Authentication will be enabled or not. It can`
	`be “true” or “false”.`
	`mail.smtp.port is the port for the SMTP service.`
	`mail.smtp.starttls.enable specifies whether TLS for SMTP will be enabled or not. It`
	`can be “true” or “false”.`
	`mail.store.protocol is the protocol to access inboxes (for email reading). Default`
	`value is “imaps”.`
	`mail.imaps.host is the address of the IMAPS server.`
	`mail.imaps.port is the port for IMAPS service.`

Credential	Credential that has been defined for this integration under the **Credential** menu.
Trust Invalid SSL Certificates	Select this if Engine's certificate certificate is self-signed or is not recognized by browsers
Require Approval From	Select users from the list who can provide approval before executing actions on this integration
Notify	Select user(s) from the list to notify when SOAR performs an action on this integration.

3. Click **Test** to test the integration.
4. Click **Save** to complete the integration.

Additional Notes

- If a SMTP integration is used without credentials then it can’t be used as incoming e-mail processor and for approvals.
- The global configuration parameter EMailDevice, under the Parameters tab of **Configuration** menu, configures the default mail server to be used in sending notifications and emails. Therefore, you must set the value of this parameter to the ID value for the SMTP Mail.

Integration Guide for Sophos XG Firewall

Integration Overview

Sophos XG Firewall is an integrated security platform featuring next gen firewall capabilities.

Integration Capabilities

ArcSight SOAR has the following integration capability with Sophos XG Firewall:

- Block IP
- Block FQDN
- Block URL
- Block Email Sender

Use Case: Blocking bad actors on firewalls

With this integration, SOAR can block malicious IP addresses, hosts and URL addresses on firewall devices while responding cyber-attacks. This can be performed automatically within a playbook or manually by an analyst.

Configuration

Prerequisites

- SOAR connects to Sophos XG Firewall API via management port. So access permission
- to this port is required.
- A user account for SOAR to connect to Sophos XG Firewall.

Configuration on Sophos XG Firewall

1. Click **Configure** > **Authentication** > **Users menu** and add an administrator user account.
2. Create a new profile or select a suitable one from the Profile list. Profile should have the following permissions:
   - Read-write for Objects
   - Web & content filter

- Email protection
- None for the rest of the permissions

3. Navigate to **Backup & Firmware** > **API** to enable API Configuration and add SOAR IP Address to the Allowed IP Address list.
4. Click **Administration** > **Device Access** to ensure that SOAR's assigned zone can access the HTTPS service of Sophos. You can prefer to create a Local Service ACL Exception Rule as well. For more information consult the Sophos How to use API documentation for further information.

1. Click **Configuration** > **Credentials** > **Create Credential**.
2. Specify the following parameter values in the **Credential Editor**:

a. **Internal Credential**

Parameter	Value
Type	Internal Credential
Name	Display name of credential set (For example, Sophos XG Credentials)
Username	Username you have created on firewall.
Password	Password you have created on firewall.
Private Key	Empty

b. **Credential Store:**

Parameter	Value
Type	External Credential
Name	Name of the credential with pull path of the safe on store

3. Click **Configuration** > **Integrations** > **Create Integration**.
4. Specify the following parameter values in the **Configuration** form:

Parameter	Value
Name	Display name of Sophos XG integration on SOAR.
Type	Sophos XG Firewall.
Address	Address of the firewall (the format should be https://192.168.10.1:4444)

Configuration	Specify the following configuration parameters:
	`# IP host group name for adding ip hosts to block`
	`iphost.group.name=ATAR_IP_BLOCK`
	`# FQDN host group name for adding fqdns to block`
	`fqdnhost.group.name=ATAR_HOST_BLOCK`
	`# Web filter url group name for adding urls to block`
	`webfilterurl.group.name=ATAR_URL_BLOCK`

Credential	Name of the credential set created on step 2 (For example, Sophos XG Credentials)
Trust Invalid SSL Certificates	Select this if Management UI's certificate certificate is self-signed or is not recognized by browsers
Require Approval From	Select users from the list who can provide approval before executing actions on this integration
Notify	Select user(s) from the list to notify when SOAR performs an action on this integration. Since SOAR only executes enrichments on Symantec DLP, leave it empty

5. Click **Test** to test the integration.
6. Click **Save** to complete the integration.

Additional Notes

- IP, FQDN and URL filter groups are automatically created by SOAR if they don't exist.1. IP, Host and URL filter groups are automatically created by SOAR if they don't exist.
- Sophos XG Firewall URL Filtering only accepts URLS with the following format http://www.example.com. URI paths are not accepted through API. Therefore SOAR transparently trim the URI part while submitting to Sophos XG Firewall.
- SOAR stores blocked email addresses in a list to keep track. Sophos currently does not provide a method to get the current list and any update will overwrite the list with the new address so administrator should only update the MTA Blocked Sender List through SOAR. Also this list is kept for each different Sophos integration but creating a second integration for the same device can lead to data inconsistency.

Integration Guide for SORBS Query

Integration Overview

SORBS Query provides free access to its DNS-based Block List to effectively block mail from more than 12 million host servers known to disseminate spam, phishing, attacks and other forms of malicious emails.

Integration Capabilities

- Action
- Check IP

Configuration

Configuration on SORBS Query

- ATAR connects to SORBS integrations's API via HTTPS. Therefore ATAR should be able to connect this service.

Configuring SOAR

Configuring SOAR

1. Click **Configuration** > **Integrations** > **Create Integration**.
2. Specify the following parameter values in the **Configuration** form:

Parameter	Value
Name	Display name of SORBS Query integration on SOAR.
Type	SORBS Query.
Address	Address of the integration (the address should be http[s]://dnsbl.sorbs.net).

Trust Invalid SSL Certificates	Select this if Engine's certificate certificate is self-signed or is not recognized by browsers
Require Approval From	Select users from the list who can provide approval before executing actions on this integration
Notify	Not Applicable

3. Click **Test** to test the integration.
4. Click **Save** to complete the integration.

Integration Guide for Splunk Enterprise Security (Alert Source)

Splunk Enterprise Security is a search for tool and can be used as both an alert source and integration. Splunk just sends webhook payload to SOAR. SOAR connects to Splunk to retrieve notable events.

SOAR can handle output from Splunk Enterprise Security in two different ways.

Incident records: SOAR connects to Splunk and gets notable events at specified time intervals. Opens incident records per notable event.

Splunk Enterprise Security runs scripts to create notable events according to defined scripts.Notable events created after the run of correlation searches. Notable events are converted to incidents in Splunk.

Notable events are transferred from Splunk to SOAR automatically at determined intervals. Contributing events are transferred to SOAR as base events.

Webhook transfers: Splunk sends data to SOAR by alerts defined at Splunk immediately using webhooks.

Alerts can be defined in Splunk's Alert Configuration. In Edit Alert trigger actions can create webhook action and when these alerts are created, Splunk sends the alert events immediately.

Configuring Splunk Enterprise Security

Address: Address of the SOAR. (the format should be http[s]://1.1.1.1:8080/api/splunk/alert).

Configuring SOAR

1. Click **Configuration** > **Integrations** > **Create Integration**.
2. Specify the following parameter values in the **Configuration** form:

Parameter	Value
Name	Display name of the alert source.
Type	Splunk Enterprise Security
Address	Address of the SOAR. (the format should be http[s]://192.168.1.10:8080/api/splunk).

Allowed IP addresses	Addresses of the allowed IPs. (the format should be http[s]://192.168.1.10:)

Configuration	Specify the following configuration parameters:
	`# Additional scope fields to be extracted from search alert`
	`(field1:CATEGORY:ROLE, field2:CATEGORY:ROLE, ...)`
	`# CATEGORY is any of: EMAIL_ADDRESS, HASH, HOST, MAC_ADDRESS, NETWORK_ADDRESS,`
	`# COMPUTER_NAME, UNKNOWN, URL, USERNAME, PROCESS`
	`# ROLE is any of: OFFENDER, IMPACT, RELATED`
	`#`
	`# Note: the fields in the example below are always extracted by default.`
	`#`
	`# Example: scope.item.fields=src_ip:NETWORK_ADDRESS:OFFENDER,`
	`dest_ip:NETWORK_ADDRESS:IMPACT, ComputerIPAddress:NETWORK_ADDRESS:RELATED,`
	`HOST:HOST:RELATED, host:HOST:RELATED, USERHOST:HOST:RELATED,`
	`dest_host:HOST:IMPACT, dest_nt_host:HOST:IMPACT, src_mac:MAC_ADDRESS:OFFENDER,`
	`mac:MAC_ADDRESS:OFFENDER, ssl_hash:HASH:RELATED, file_hash:HASH:RELATED,`
	`url:URL:RELATED, UserName:USERNAME:RELATED,`
	`OS_USERNAME:USERNAME:RELATED, DBUSERNAME:USERNAME:RELATED,`
	`process_id:PROCESS:RELATED, IDProcess:PROCESS:RELATED`
	`# scope.item.fields=`
	`# configure how far (in minutes) into the past this enrichment will look.`
	`#cache.reusing.duration=20`
	`Note: the fields in the example below are always extracted by default.`
	`scope.item.fields=src_ip:NETWORK_ADDRESS:OFFENDER,`
	`dest_ip:NETWORK_ADDRESS:IMPACT,`
	`ComputerIPAddress:NETWORK_ADDRESS:RELATED,`
	`HOST:HOST:RELATED,`
	`host:HOST:RELATED,`
	`USERHOST:HOST:RELATED,`
	`dest_host:HOST:IMPACT,`

	`dest_nt_host:HOST:IMPACT,`
	`src_mac:MAC_ADDRESS:OFFENDER,`
	`mac:MAC_ADDRESS:OFFENDER,`
	`ssl_hash:HASH:RELATED,`
	`file_hash:HASH:RELATED,`
	`url:URL:RELATED,`
	`UserName:USERNAME:RELATED,`
	`OS_USERNAME:USERNAME:RELATED,`
	`DBUSERNAME:USERNAME:RELATED,`
	`process_id:PROCESS:RELATED,`
	`IDProcess:PROCESS:RELATED`
	`scope.item.fields=`

Credential	Credential that has been defined for this alert source under the **Credentials** menu.
Trust Invalid SSL Certificates	Select this if Engine's certificate certificate is self-signed or is not recognized by browsers

3. Click **Test** to test the integration.
4. Click **Save** to complete the integration.

SOAR Configuration Parameters

- SplunkBaseEventCountPerRequest Maximum number of base events to retrieve in a single API request Integer 1000
- SplunkListenerAutoEnrichEnabled Enable Splunk listener auto-enrichment with base-event data Boolean false
- SplunkNotableEventCountPerRequest Maximum number of notable events to retrieve in a single API request Integer 1000
- SplunkSyncPeriod Period in seconds to sync Splunk events Integer 60

```
incident-repoened = true (default)
```

```
auto-close = true (default)
```

```
try-period (hour)
```

Additional Notes

- In webhook configuration api/splunk/alert must be added to the end of SOAR address as stated before.
- When incidents that are closed in Splunk are also transferred to SOAR automatically. We can close Splunk incidents through SOAR manually by action plugin.

```
incident.autoReopen = true
```

```
incident.autoClose = false
```

```
incident.autoSync = false
```

```
incident.autoSyncRetryPeriod = 1 (if notable event's enrichment is
unsuccessful
```

```
(which means base events are not received), period for retry (default 1
hour)
```

- **Configuration** > **Parameters** > **SplunkSyncPeriod** (default 60 seconds)

Integration Guide for Splunk Enterprise Security

Integration Capabilities

- Action

Configuring SOAR

1. Click **Configuration** > **Integrations** > **Create Integration**.
2. Specify the following parameter values in the **Configuration** form:

Parameter	Value
Name	Display name of Splunk Enterprise Security integration on SOAR
Type	Splunk Enterprise Security
Address	Address of the integration (the format must be https://1.1.1.1:1234 https://abc.example.com:1234).
Credential	Credential that has been defined for this alert source under the **Credentials** menu.
Trust Invalid SSL Certificates	Select this if Engine's certificate certificate is self-signed or is not recognized by browsers
Require Approval From	Select user(s) from list to ask her/his approval before executing actions on this integration.
Notify	Select user(s) from the list to notify when SOAR performs an action on this integration.

3. Click **Test** to test the integration.
4. Click **Save** to complete the integration.

Integration Guide for Symantec Advanced Threat Protection

Integration Overview

Symantec Advanced Threat Protection is Symantec's endpoint protection platform closely works with SEP Manager.

Integration Capabilities

- Action Capabilities
- Quarantine Endpoint (isolate_endpoint and rejoin_endpoint)
- Delete File From Endpoint (delete_endpoint_file)
- Enrichment Capabilities
- Get Events (/events)

Configuration

Configuring Symantec Advanced Threat Protection

Symantec ATP uses https (tcp/443) for API access by default.

1. Click **Settings** > **Data Sharing** > **OAuth Clients** > **Add application with custo role** to add the API application.
2. The image in the **Privileges** section represents how the custom role must be configured. After creating user, Symantec displays the **client secret** and **client id**, which is used in SOAR configuration modal.

Configuring SOAR

1. Navgate to **Configuration** > **Integrations**.
2. Specify the following parameter values in the **Integrations Editor**:

Parameter	Value
Name	Display name of Symantec Advanced Threat Protection integration on SOAR
Type	Symantec Advanced Threat Protection.
Address	Address of the integration (in the following format: https://1.1.1.1)
Configuration	Specify the following configuration parameters. `#EVENT_RESULT_LIMIT`
Credential	Name of the credential set created under the **Credentials** menu. You must use **client id** as username and **client secret** as password.
Trust Invalid SSL Certificates	Select this if Engine's certificate is self-signed or is not recognized by browsers
Require Approval From	Select users from list who can provide approval before executing actions on this integration
Notify	Select users from the list to notify when SOAR performs an action on this integration

3. Click **Test** to test the integration.
4. Click **Save** to save the integration.

Integration Guide for Symantec Bluecoat Malware Analysis Appliance (MAA)

Integration Overview

Symantec Bluecoat MAA is a malware analyzer sand-box solution. SOAR uses Symantec Bluecoat Malware Analysis Appliance to analyze files and URLs.

Integration Capabilities

- Action
- File Analysis
- Hash Analysis
- URL Analysis

Prerequisites

- SOAR connects to Symantec Bluecoat MAA's Remote API (RAPI) via HTTPS. Therefore, SOAR should be able to connect this service.
- A user credential with admin role and its token are required.

Configuration

Configuring Symantec Bluecoat Malware Analysis Appliance (MAA)

- To generate a token, click **System Settings** > **Users** > **Add New API Key** on the appliance.

Configuring SOAR

1. Navgate to **Configuration** > **Integrations**.
2. Specify the following parameter values in the **Integrations Editor**:

Parameter	Value
Name	Display name of Symantec Bluecoat MAA integration on SOAR.
Type	Symantec Bluecoat MAA .
Address	Address of the integration (in the following format: http[s]://1.1.1.1:1234
Credential	Name of the credential set created under the **Credentials** menu. The user's API token should be set as the password while creating integration credential.
Trust Invalid SSL Certificates	Select this if Engine's certificate is self-signed or is not recognized by browsers
Require Approval From	Select users from list who can provide approval before executing actions on this integration
Notify	Select users from the list to notify when SOAR performs an action on this integration

3. Click **Test** to test the integration.
4. Click **Save** to save the integration.

Integration Guide for Symantec BlueCoat Proxy SG

Integration Overview

BlueCoat Proxy SG is a secure web gateway solution developed by Symantec which controls the users' access to web content.

Integration Capabilities

ATAR has the following integration capability with Symantec BlueCoat Proxy SG

- Block

Use Case: Blocking access to malicious URL

ATAR can integrate with Symantec BlueCoat Proxy SG to block malicious URLs detected while responding an incident. Blocking can be performed automatically within a playbook or manually by an analyst.

Configuration

Prerequisites

- Currently SOAR supports Symantec BlueCoat Proxy SG version 6.6.4.2 and connects to Symantec BlueCoat Proxy SG Management UI through HTTPS in order to download existing copy of local database. As Management Console runs on 8082 /tcp port, so access to this port is required.
- SOAR connects to Symantec BlueCoat Proxy SG via SSH to immediate update of local database. So access to 22/tcp port is required.
- Symantec BlueCoat Proxy SG connects back to SOAR API to gather new copy of the local database. As SOAR API runs on 443/tcp port, so access from BlueCoat Proxy SG to this service is required.
- Admin user credentials are required for SOAR to connect Symantec BlueCoat Proxy SG

Configuring Symantec BlueCoat Proxy SG

1. Click **Configuration** > **Content Filtering** > **General** and enable **Local Database**.
2. Click **Configuration** > **Content Filtering** > **Local Database** and configure copy of local database URL accessible on SOAR . The format should be `https://atar/api/bluecoat/list/{integrationId}`

   `integrationId`: ID of BlueCoat Proxy SG integration on SOAR.

Configuring SOAR

1. Click **Configuration** > **Credentials** > **Create Credential**.
2. Specify the following parameter values in the **Credential Editor**:
   a. **Internal Credential**

Parameter	Value
Type	Internal Credential
Name	Display name of credential set (For example, BlueCoat Proxy SG Credentials)
Username	Username of the administrator
Password	Password of the admin user
Private Key	Empty

   b. **Credential Store:**

Parameter	Value
Type	External Credential
Name	Name of the credential with pull path of the safe on store

3. Click **Configuration** > **Integrations** > **Create Integration**.
4. Specify the following parameter values in the **Configuration** form:

Parameter	Value
Name	Display name of Symantec BlueCoat Proxy SG integration on SOAR
Type	Symantec BlueCoat Proxy SG
Address	Address of the integration (in the following format: https://192.168.2.99)

Configuration	Specify the following configuration parameters:
	`# Default category to block URLs. If empty, value of`
	`# BlueCoatDefaultBlockListCategoryName configuration`
	`# parameter will be used.`
	`#category=atar`
	`# Comma (,) separated list of IP addresses of Bluecoat`
	`# servers that are allowed to retrieve blocked URL list.`
	`# If unspecified, specified device address will be used.`
	`#allowedaddresses=`
	`# Default block list source URL. This URL should be pointed out`
	`# third-party block list source address. If unspecified, value`
	`# of BlueCoatDefaultBlockListURL will be used.`
	`#blocklistsource=`
	`# Connect to Bluecoat Proxy using SSH with provided`
	`# credential and execute commands to immediately force`
	`# refresh of the block list. Default is false.`
	`#forcerefresh.enabled=false`
	For a third party blacklist to work correctly it must be structured as follows:
	For example,
	If you want to work with seperate categories you can give a different category name to differentiate between SOAR sourced URL's and the third-party URL's.
	`define category "atar"`
	`www.example.com`
	`www.example.com/example.asp`
	`example.com`
	`192.168.201.57`
	`end category "atar"`

Credential	Name of the credential set created on step 2 (For example, BlueCoat Proxy SG Credentials)

Trust Invalid SSL Certificates	Select this if Management Consoles's certificate is self-signed or is not recognized by browsers
Require Approval From	Select users from the list who can provide approval before executing actions on this integration
Notify	Not Applicable

5. Click **Test** to test the integration.
6. Click **Save** to complete the integration.

Additional Notes

- Due to update mechanism of Blucoat Proxy SG's Content Filter/Local Database, BlueCoat Proxy SG retrieves the list of items to be blocked from a URL located on a web server that is accessible by the Proxy SG. SOAR maintains a copy of Content Filter/Local Database and is accessible on https://atar/api/bluecoat/list/{integrationId}.
- SOAR connects to management console and downloads a copy of the Content Filter/Local Database before adding new entries. If SOAR is the only place managing Content Filter/Local Database, you don't need to provide this access since ATAR always has the latest copy.
- After updating the list of items to be blocked on itself, SOAR might connect to BlueCoat Proxy SG via SSH and trigger an immediate download of the Content Filter/ Local Database file. This operation requires to access privileged-mode. In order to use this method set `forcerefresh.enabled=true` on integration configuration. List of commands executed during this operation can be found under **Configuration** > **Customization Library** > **Symantec BlueCoat Proxy SG SSH Integration Action** (Block) Default Template.
- If **Automatically check for updates** is set on Content Filter/Local Database configuration BlueCoat periodically connects and checks the latest version of the list. If you don't want immediate update you may set `forcerefresh.enabled=false` on integration configuration and prefer to use automatic updates.

Integration Guide for Symantec Bluecoat Site Review

Integration Overview

Bluecoat Site Review is a site to report uncategorized URLs to Symantec/Bluecoat.

Integration Capabilities

- Action
- Report Uncategorized URL (should get URL from scope)

Configuration

Configuration on Bluecoat Site Review

No requirements

Configuring SOAR

- In SOAR **Configuration**, specify **Name**, **Address** and **submissionEmailAddress** to check submission result from returning mail.

Note: Add a dummy credential that can be removed in future releases.

Integration Guide for Symantec Data Loss Prevention (DLP)

Integration Overview

Symantec DLP is a solution to ensure that sensitive data is not lost, misused, or accessed by unauthorized users

Integration Capabilities

SOAR has the following integration capabilities with Symantec DLP:

- Retrieve incidents

Use Case: Investigating Suspicious Behaviour

During investigation of a suspicious behaviour of an employee or an endpoint, SOAR integrated with Symantec DLP, can get access the related DLP incidents for better understanding of the case. Investigation can be performed automatically within a playbook or manually by an analyst.

Configuration

Prerequisites

- Currently SOAR supports Symantec DLP 14.6.0200 version. SOAR connects to Symantec DLP API via HTTPS. Access to 443/tcp port is required.
- A user account is required for SOAR to connect to Symantec DLP.

Configuring Symantec DLP

1. Login to Symantec DLP Enforce Server and navigate to **System** > **Login Management** > **Roles** to create a web service role. The web service role should have the following permissions:
   - Incidents: View
   - Perform Attribute Lookup
   - Incident Reporting and Update API: Incident Reporting

- Display Attributes: All,
- Custom Attributes: View all

2. Click **System** > **Login Management** > **DLP Users** and add a DLP user account with the role that is created on previous step.
3. Login to Symantec DLP Enforce server administration console with the DLP user account created in previous step.
4. Click **Incidents** > **Incident Reports** and select a system defined incident list, such as **Incidents - All**.
5. Edit report filters to narrow down the results to be returned if needed. In the **Summarize by** menu verify that **and** are both selected.
6. Save the report as a new private report and note the new report's ID.

Configuring SOAR

1. Click **Configuration** > **Credentials** > **Create Credential**.
2. Specify the following parameter values in the **Credential Editor**:
   a. **Internal Credential**

Parameter	Value
Type	Internal Credential
Name	Display name of credential set (For example, Symantec DLP Credentials)
Username	User you have created for SOAR on Symantec DLP.
Password	Password of the user you have created for SOAR on Symantec DLP
Private Key	Empty

   b. **Credential Store:**

Parameter	Value
Type	External Credential
Name	Name of the credential with pull path of the safe on store

3. Click **Configuration** > **Integrations** > **Create Integration**.
4. Specify the following parameter values in the **Configuration** form:

Parameter	Value
Name	Display name of Symantec DLP integration on SOAR.

Type	Symantec Data Loss Prevention.
Address	Address of the integration (in the following format: https://192.168.2.15)

Configuration	Specify the following configuration parameters:
	`# Report id`
	`report.id=221`

Credential	Name of the credential set created on step 2 (For example, Symantec DLP Credentials)
Trust Invalid SSL Certificates	Select this if Web UI's certificate certificate is self-signed or is not recognized by browsers
Require Approval From	Select users from the list who can provide approval before executing actions on this integration
Notify	Select user(s) from the list to notify when SOAR performs an action on this integration. Since SOAR only executes enrichments on Symantec DLP, leave it empty

5. Click **Test** to test the integration.
6. Click **Save** to complete the integration.

Additional Notes

For the details of web service role and report creation please refer to Symantec™ Data Loss Prevention Incident Reporting and Update API Developers Guide.

Integration Guide for Symantec DeepSight Intelligence

Integration Overview

Symantec DeepSight Intelligence is a commercial threat intelligence service which provides actionable intelligence with context and technical details surrounding a threat so teams can quickly assess cyber risk and implement proactive controls.

Integration Capabilities

ArcSight SOAR has the following integration capabilities with Symantec DeepSight Intelligence Service:

- Ingest intelligence data as alert
- Domain Query
- File Query
- IP Query

Use Case: Investigating Phishing Campaigns

SOAR is integrated with Symantec DeepSight Intelligence, to help investigation and mitigation of phishing campaigns. When a phishing report email comes from user, SOAR extracts the indicators such as IP address, domains and attachments in message and a new incident is created on SOAR's own Incident Management Service Desk. SOAR then asks these indicators to Symantec DeepSight Intelligence if this is a known attack and previously analyzed. This can be performed automatically within a playbook or manually by an analyst.

Configuration

Prerequisites

- SOAR connects to Symantec DeepSight API via HTTPS. Access to https://deepsightapi.symantec.com/v1/ (443/tcp port) and https://datafeeds.symantec.com/ (443/tcp port) is required.
- A user account and a certificate-password pair are required for SOAR to connect to Symantec DeepSight. These will be supplied by Symantec through DeepSight portal.

Configuring Symantec DeepSight Intelligence

SOAR requires a username and password to be created on Symantec DeepSight for authentication purposes for Alert Source. If enrichment capabilities are to be used an API key must be enabled and created.Use an administrator account to enable API Access for the account you wish to use in SOAR.

1. Select **user's detail** tab. The tab includes a section for DeepSight API Token. Select **Enable Access**
2. Login with the SOAR account to the DeepSight portal.
3. Click **Settings** > **My Profile** and locate the **DeepSight API Token** tab.
4. Copy the API key.

Configuring SOAR

1. Click **Configuration** > **Credentials** > **Create Credential**.
2. Specify the following parameter values in the **Credential Editor**:
   a. **Internal Credential**

Parameter	Value
Type	Internal Credential
Name	Display name of credential set (For example, Symantec DeepSight Credentials).
Username	Empty
Password	API Key you've get from Symantec DeepSight Intelligence platform.
Private Key	Empty

   b. **Credential Store:**

Parameter	Value
Type	External Credential
Name	Name of the credential with pull path of the safe on store

Configuring Symantec DeepSight Intelligence as Alert Source

1. Click **Configuration** > **Alert Source** > C**reate Alert Source Configuration**.
2. Specify the following parameter values in the **Configuration** form:

Parameter	Value
Name	Display name of Symantec DeepSight Intelligence Alert Source on SOAR.
Type	Symantec DeepSight Intelligence Datafeeds
Address	Address of the Symantec DeepSight Intelligence DataFeeds (https://datafeeds.symantec.com/v1/).

Configuration	Specify the following configuration parameters:
	`# Number of item to ingest per data feed type on first integration`
	`alertCountPerFeedType=1000`
	`# Minimum item reputation value to turn into Alert on SOAR`
	`minReputationToAlert=10`
	`#usable behaviour names : attack,attacks,bot,cnc,fraud,malware,phish,spam,phish_host`
	`#behaviourNames=attack,bot,CnC,fraud,malware,spam`
	`# Integration ID of the proxy integration to use when connecting to current source.`
	`# If not provided, SOAR will try to use a direct connection.`
	`#proxy.id=5422`
	`# configure how far (in minutes) into the past this enrichment will look.`
	`#cache.reusing.duration=20`

Credential	Name of the credential set created on step 2 (For example, Symantec DeepSight Credentials)
Trust Invalid SSL Certificates	Select this if Web UI's certificate certificate is self-signed or is not recognized by browsers
Require Approval From	Select users from the list who can provide approval before executing actions on this integration. Since SOAR only executes enrichment on Symantec DeepSight, leave it empty.
Visible Alert Fields	You may define which alarm fields will be displayed on Incident Management Service Desk.

Configuring Symentec DeepSight Intelligence as Integration

1. Click **Configuration** > **Alert Source** > C**reate Alert Source Configuration**.
2. Specify the following parameter values in the **Configuration** form:

Parameter	Value
Name	Display name of Symantec DeepSight Cyber Intelligence integration on SOAR.
Type	Symantec DeepSight Cyber Intelligence
Address	Address of Symantec DeepSight Cyber Intelligence (https://deepsightapi. symantec.com/v1)

Configuration	Specify the following configuration parameters:
	`# Integration ID of the proxy integration to use when connecting to current integration.`
	`# If not provided, SOAR will try to use a direct connection.`
	`#proxy.id=123`
	`# configure how far (in minutes) into the past this enrichment will look.`
	`#cache.reusing.duration=20`

Credential	Name of the credential set created on step 2 (For example, Symantec DeepSight Credentials)
Trust Invalid SSL Certificates	Select this if Web UI's certificate certificate is self-signed or is not recognized by browsers
Require Approval From	Select users from the list who can provide approval before executing actions on this integration. Since SOAR only executes enrichment on Symantec DeepSight, leave it empty.
Notify	Select users from the list to notify when SOAR performs an action on this integration. Since SOAR only executes enrichment on Symantec DeepSight, leave it empty.

3. Click **Test** to test the integration.
4. Click **Save** to complete the integration

Integration Guide for Symantec Endpoint Protection Manager

Integration Overview

Symantec Endpoint Protection Manager (SEP Manager) is a management platform for security software suite, which consists of anti-malware, intrusion prevention and firewall features for server and desktop computers.

Integration Capabilities

SOAR has the following integration capabilities with Symantec Endpoint Protection Manager:

- Start Scan on Client
- Block File Hash
- Get Client Info

Use Case: Starting scan jobs on suspicious endpoints.

During the course of and investigation or responding a ongoing cyber-attack, it is required to run scan jobs on suspicious endpoints to validate the threat. SOAR can start scan jobs on Symantec Endpoint Protection Manager to help on deciding the next course of action.

This can be performed automatically within a playbook or manually by an analyst.

Configuration

Prerequisites

- Currently SOAR supports Symantec Endpoint Protection Manager 14.2.760 version. SOAR needs to connect Symantec Endpoint Protection Manager API and Database.
- Access to 8443/tcp, 8446/tcp port for API acceess and 1433/tcp, 1434/udp port for database access is required.
- User accounts for API access and database access are required for SOAR to connect to Symantec Endpoint Protection Manager.

Configuring Symantec Endpoint Protection Manager

1. Login to SEP Management Server on https://SEPManager:8443/console/apps/sepm and create an administrator account on **Admin** tab.
2. Click **Policy** > **Policy Components > File Fingerprint Lists** and add a File Fingerprint List.
3. You might create a file containing MD5 value of eicar.com test signature 44d88612fea8a8f36de82e1278abb02f: to upload a file to create the list.
4. Login to SEP Manager Web Service Application Registration on https://SEPManager:8443/sepm with the admin account you've created on previous step and register a webservice application to be used by SOAR.

   Note the Client ID and Client Secret values are generated.

5. Create a database user that has selected permissions and ensure that the SQL Browser service is configured and running on MSSQL Server.

Configuring SOAR

1. Click **Configuration** > **Credentials** > **Create Credential**.
2. Specify the following parameter values in the **Credential Editor**:
   a. **Internal Credential**

Parameter	Value
Type	Internal Credential
Name	Display name of credential set (For example, SEP Manager Credentials).
Username	Username you have created for SOAR on Symantec Endpoint Protection Manager
Password	Password of the user you have created for ATAR on Symantec Endpoint Protection Manager.
Private Key	Empty

   b. **Credential Store:**

Parameter	Value
Type	External Credential
Name	Name of the credential with pull path of the safe on store

3. To create credentials to be used for database connection:

a. **Internal Credential**

Parameter	Value
Type	Internal Credential
Name	Display name of credential set (For example, SEP Manager DB Credentials).
Username	Database username you have created for SOAR on SEP Manager Database.
Password	Password of the user you have created for SOAR on SEP Manager Database.
Private Key	Empty

b. **Credential Store:**

Parameter	Value
Type	External Credential
Name	Name of the credential with pull path of the safe on store

4. Click **Configuration** > **Integrations** > **Create Integration**.
5. Specify the following parameter values in the **Configuration** form:

Parameter	Value
Name	Display name of Symantec Endpoint Protection Manager integration on SOAR
Type	Symantec Endpoint Protection Manager
Address	Address of the integration (in the following format: https://192.168.2.140)

Configuration	Specify the following configuration parameters:
	`client.id=xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx`
	`client.secret=xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx`
	`#domainName=`
	`directdbaccess.enabled=true`
	`directdbaccess.jdbcurl=`
	`jdbc:sqlserver://192.168.2.140:1433\\SEPMDB;database=sem5`
	`directdbaccess.credential=33323`
	`# Integration ID of the proxy integration to use when connecting to`
	`# current integration.`
	`# If not provided, ATAR will try to use a direct connection.`
	`#proxy.id=123`

Credential	Name of the credential set created on step 2 (For example, SEP Manager Credentials).
Trust Invalid SSL Certificates	Select this if Engine's certificate is self-signed or is not recognized by browsers
Require Approval From	Select users from the list who can provide approval before executing actions on this integration
Notify	Select user(s) from the list to notify when ATAR performs an action on this integration.

6. Click **Test** to test the integration.
7. Click **Save** to complete the integration.

Additional Notes

Symantec Endpoint Protection Manager Webservice registration works on 8446/tcp port by default. If it is different than this value, you might configure it using **DefaultSEPMRestApiPort** paramater under **Configuration** > **Parameters**.

Integration Guide for Symantec Managed Security Services (MSS)

Integration Overview

Symantec Managed Security Services (MSS) provides its customers security monitoring and real-time security analytics services including strategic insights needed to prioritize and respond to incidents and build strategies to protect the assets, reputations and viability of their organizations.

Integration Capabilities

SOAR has the following integration capabilities with Symantec MSS:

- Ingest Incident Records as Alert
- Update MSS incident record
- Close MSS incident

Use Case #1: Investigating and Mitigating Cyber-attacks

Integrated with Symantec MSS, ATAR periodically collects new incidents and update the statuses of the open incidents as they change in Smyantec MSS system. When an incident record is created on Symantec MSS, ATAR automatically collects Incident Details such as Analyst Comment, Signatures that are triggering this alert, Comments that are added to the incident and possible Attachments inside this alert anc creates a new incident on its own Incident Management Service Desk.

Configuration

Prerequisites

- SOAR connects to Symantec MSS API via HTTPS. So access permission to https://api.

managedsecurity.com is required.

- A user account and a certificate-password pair are required for ATAR to connect to Symantec MSS API.

Configuring Symantec MSS

The Symantec MSS service uses client-side certificates for authentication.

1. Click **Profile** > **Certificates** > **Create a certificate.**
2. Select the **type of service** for the certificate.
3. Set the expiration date for the certificate. The available values are 6 months, 1 year, and 2 years.
4. [Optional] Specify the name for the certificate.
5. Click **Register**.

> The certificates are enabled by default upon creation, but must be downloaded and installed before they can be used.

Configuring SOAR

To use the client-side certificate created on Symantec MSS, you must convert it with **openssl** command line tool as following:

```
openssl pkcs12 -in <certificate_created_in_MSS_Portal>.p12 -clcerts -nodes -
out <output_file>
```

Configuring Credentials

1. Click **Configuration** > **Credentials** > **Create Credential**.
2. Specify the following parameter values in the **Credential Editor**:
3. **Internal Credential**

Parameter	Value
Type	Internal Credential
Name	Display name of credential set (For example, Symantec MSS Credentials)).
Username	Empty
Password	Empty.
Private Key	Paste the content of the <output_file>.pem file into the Private Key area.

> The external credential stores can not be used with this integration type.

Configuring Symantec MSS as Alert Source

1. To add a new incident severity configuration, click **Configuration** > **Incidents** > **Severities** . Symantec MSS integration requires the following incident severity definitions:
   - Informational
   - Warning
   - Critical
   - Emergency
2. To add a new incident statues configuration, click **Configuration** > **Incidents** > **Statuses**. Symantec MSS integration requires the following incident status definitions:
   - New
   - In Progress as Open statuses
   - False Positive
   - Resolved
   - Deferred
   - No Action as closed statuses.
3. Click **Configuration** > **Alert Source** > **Create Alert Source Configuration.**
4. Specify the following parameter values in the **Configuration** form:

Parameter	**Value**
Name	Display name of Symantec MSS Alert Source on SOAR
Type	Symantec MSS
Address	Address of Symantec MSS service (in the following format: https://api.monitoredsecurity.com).
Alert Severities	Mapping of alert severity values to SOAR incident severities.

Configuration	Specify the following configuration parameters: # Enables incident sync # Default: false #incident.autoSync=true # Request timeout in minutes # If not provided, ATAR will use 10 by default #request.timeout=10 # Enable auto closing ATAR incidents when the related Symantec MSS incident is closed, # Default: false #incident.autoClose=true # Enable auto reopening ATAR incidents when the related Symantec MSS incident is reopened, # Default: false #incident.autoReopen=true # Scope fields to be extracted from base events and/or correlated events (field1:CATEGORY:ROLE, # CATEGORY is any of: EMAIL_ADDRESS, HASH, HOST, MAC_ADDRESS, NETWORK_ADDRESS, # COMPUTER_NAME, UNKNOWN, URL, USERNAME, PROCESS # ROLE is any of: OFFENDER, IMPACT, RELATED # # Note: The fields in the baseevent.scope example below are always extracted by default. # Note: Extraction with same field name overrides the default one. # Note: Extraction with different field name does not override the default behaviour and extracted # Note: Field names must start with / character # # Example: baseevent.scope=/sourceIPString:NETWORK_ADDRESS:OFFENDER # baseevent.scope= # # Example: correlated.scope=/sourcev6:NETWORK_ADDRESS:OFFENDER # correlated.scope= # How far (in days) into the past ATAR will look for remote incidents at the initial sync task

	`# If not provided, ATAR will use 14 days by default` `#days.to.look.back.at.initial.sync=14`
Credential	Name of the credential set you have created (For example, Symantec MSS Credentials).
Trust Invalid SSL Certificates	Select this if Engine's certificate is self-signed or is not recognized by browsers
Viisible Alert Fields	Select alarm fields that has to be displayed on Incident Management Service Desk.
Notify	Select user(s) from the list to notify when ATAR performs an action on this integration.

5. Click **Test** to test the integration.
6. Click **Save** to complete the integration.

Configuring Symantec MSS as an Integration

1. Click **Configuration** > **Integrations** > **Create Integration**.
2. Specify the following parameter values in the **Configuration** form:

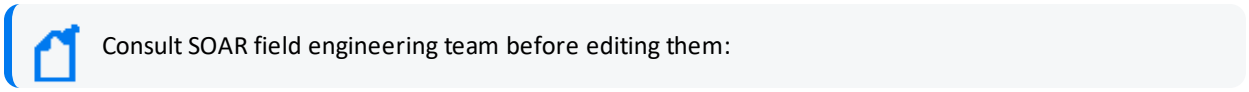
Parameter	Value
Name	Display name of Symantec MSS integration on SOAR
Type	Symantec MSS
Address	Address of Symantec MSS service (in the following format: https://api.monitoredsecurity.com).
Configuration	Specify the following configuration parameters: `#proxy.id=5422`
Credential	Name of the credential set you have created (For example, ArcSight ESM Credentials).

Trust Invalid SSL Certificates	Select this if Engine's certificate is self-signed or is not recognized by browsers
Require Approval From	Select user(s) from list to ask her/his approval before executing actions on this integration.
Notify	Select user(s) from the list to notify when ATAR performs an action on this integration.

3. Click **Test** to test the integration.
4. Click **Save** to complete the integration.

Additional Notes

The following configuration parameters can be used for fine tuning the integration.



Parameter Name Description Default Value

```
SymantecMssListenerMaxRetrySeconds Symantec MSS listener queue max message retry in seconds 1800
```

```
SymantecMssListenerQueueConcurrency Upper limit of Symantec MSS Listener consumer thread count 3
```

```
SymantecMssSyncLookBehindMinutes Minutes to look behind to incident in Symantec MSS SyncTask 20
```

```
SymantecMssSyncPeriod Period in seconds to sync Symantec MSS incidents 60
```

```
Below Automation Bit sample could be used to automatically close incidents via Trigger.
```

```
atar.require("underscore");
```

```
var remoteStatusList = [
```

```
'False Positive',
```

```
'Resolved',
```

```
'Deferred',
```

```
'No Action'
```

```
];
```

```
var remoteStatus = 'Resolved';
```

```
var statusName = atar.getTicket().getTicketStatus().getName();
if (_.contains(remoteStatusList, statusName)) {
remoteStatus = statusName;
}
var params = {'INCIDENT_CLOSING_STATUS': remoteStatus};
atar.action(ActionPluginCapability.CLOSE_INCIDENT, atar.getAlert(),
atar.device("Symantec MSS Integration"), params);
```

Integration Guide for Symantec Messaging Gateway

Integration Overview

Symantec Messaging Gateway (Brightmail) is an email gateway which is used to filter incoming and outgoing emails.

Integration Capabilities

SOAR has the following integration capabilities with Symantec Messaging Gateway:

- Block Sender
- Block in Dictionary

Use Case: Blocking phishing attacks

SOAR can follow the email inboxes for user's phishing reports and automatically creates an incident record on its service desk. To stop the phishing campaigns, SOAR can extract the sender address, IP, e-mail subject and block them on Symantec Messaging Gateway.

This can be performed automatically within a playbook or manually by an analyst.

Configuration

Prerequisites

- Currently SOAR supports Symantec Messaging Gateway 10.6.5-1 version.
- SOAR connects to Symantec Messaging Gateway via HTTPS. Access to 443/tcp port is required.
- A user account for SOAR to connect Symantec Messaging Gateway.

Configuring Symantec Messaging Gateway

1. Click **Administration** > **Users** and select **Create a new administration policy** to create an administrator account. Select **Manage Policies right**.

   Disable all other rights since they are unnecessary.
2. Click **Content** > **Dictionaries** to create a dictionary.

3. To block hosts and IP addresses, SOAR uses **Local Bad Sender IPs** and**Local Bad Sender Domains**.

Configuring SOAR

1. Navigate to **Configuration** > **Credentials** and click **Create Credential.**
2. Fill the **Credential Editor** form with following parameter values:
   a. **Internal Credential:**

Parameter	Value
Type	Internal Credential
Name	Display name of credential set (For example, Symantec Messaging Gateway Credential)
Username	Username you have created of SOAR on Symantec Messaging Gateway
Password	Password of the user you have created of SOAR on Symantec Messaging Gateway.
Private Key	Empty

   b. Credential Store

Parameter	Value
Type	External Credential
Name	Name of the credential with full path of the safe on store

3. Click **Configuration** > **Integrations** > **Create Integration**.
4. Fill the configuration form with the following parameter values:

Parameter	Value
Name	Display name of Symantec Messaging Gateway integration on SOAR.
Type	Symantec Messaging Gateway.
Address	Address of the integration (the format must be192.168.2.212).)
Configuration	You need to specify the following configuration parameters. You can define multiple dictionaries by seperating "|", for example, dictionary.name=SOAR Dictionary 1 | SOAR Dictionary 2
Credential	Name of the credential set you've just created on step 2 (for example, Symantec Messaging Credential.

Trust Invalid SSL Certificates	Select this if Symantec Messaging Gateway's certificate is self-signed or not recognized by browserss.
Require Approval From	Select user(s) from list to ask her/his approval before executing actions on this integration.
Notify	Select user(s) from the list to notify when SOAR performs an action on this integration.

5. Click **Test** to test the integration.
6. Click **Save** to complete integration.

Integration Guide for Tenable Nessus

Integration Overview

Tenable Nessus is a vulnerability scanner used to detect vulnerabilities on the network. SOAR uses Tenable Nessus to gather vulnerability information to enrich incidents' context.

Integration Capabilities

- Action
- Get Scan List
- Get All Vulnerabilities on a Scan

Configuration

ConfiguringTenable Nessus

- SOAR connects to Tenable Nessus' API via HTTPS. Therefore SOAR must be able to connect this service.
- A user credential is required.

Configuration on SOAR

Configuring SOAR

1. Navigate to **Configuration** > **Integrations**.
2. In the **Integrations Editor**, specify the following parameter values:

Parameter	Value
Name	Display name of Tenable Nessus integration on SOAR
Type	Tenable Nessus.
Address	Address of the integration (in the following format: http[s]://1.1.1.1:1234 or http [s]://abc.example.com:1234
Credential	Credential defined for the integration under the **Credentials** menu

Trust Invalid SSL Certificates	Select this if Engine's certificate is self-signed or is not recognized by browsers
Require Approval From	Select users from the list who can provide approval before executing actions on this integration
Notify	Select users from the list to notify when SOAR performs an action on this integration

3. Click **Test** to test the integration.
4. Click **Save** to complete the integration.

Integration Guide for Tenable Security Center

Integration Overview

Tenable Security Center (Tenable SC) is a vulnerability management solution that provides visibility into network by identifying all vulnerabilities, misconfigurations and malware attack on assets and gives ability to manage and measure your cyber risk.

SOAR has the following integration capabilities with Tenable Security Center:

- Get Assets
- Get Vulnerabilities (System-wide)
- Get Vulnerabilities on IP

.**Use Case: Getting vulnerability details of assets**

SOAR can integrate with Tenable Security Center to gather additional information about an asset during incident investigatio. Knowing existing vulnerabilities on a system can help SOC analysts to understand possible root cause of an incident more precisely.

Configuration

Prerequisites

- SOAR connects to Tenable Security Center's API using HTTPS. Typically an access permission to 443/tcp port is required.
- A user account for SOAR to connect to Tenable Security Center.

Configuring Tenable Security Center

1. Login to Tenable Security Center with Security Manager User.

   Note: This user account is different from admin account.

2. Navigate to **Users**> **Groups** and add a group to define the objects that SOAR can access. You must at select atleast one item from **Viewable Hosts and Repositories lists**.

   There is no need to share any object under **Share to Group** tab.

3. To add user for SOAR access, navigate to **Users** > **Users**. Select **No Role** and **SOAR Access Group** in **Membership**.

Configuring SOAR

1. Navigate to **Configuration** > **Credentials** and click **Create Credential.**
2. Fill the **Credential Editor** form with following parameter values:
   a. **Internal Credential:**

Parameter	Value
Type	Internal Credential
Name	Display name of credential set (For example, Tenable SC Credential
Username	User you have created of SOAR on Tenable Security Center.
Password	Password of the user you have created of SOAR on Tenable Security Center.
Private Key	Empty

3. Click **Configuration** > **Integrations** > **Create Integration**.
4. Fill the configuration form with the following parameter values:

Parameter	Value
Name	Display name of Tenable Security Center integration on SOAR.
Type	Tenable Security Center.
Address	Address of the integration (the format must be https://1.1.1.1:1234 or https://abc.example.com:1234)
Credential	Name of the credential set you've just created on step 2 (for example, Tenable SC Credential.
Trust Invalid SSL Certificates	Select this if Engine's certificate is self-signed or not recognized by browserss.
Require Approval From	Select user(s) from list to ask her/his approval before executing actions on this integration.
Notify	elect user(s) from the list to notify when SOAR performs an action on this integration.

5. Click **Test** to test the integration.
6. Click **Save** to complete integration.

Integration Guide for Trend Micro Control Manager

Integration Overview

Trend Micro Control Manager is a centralized management server for Trend Micro products to protect endpoints from malware and network threats. SOAR uses Trend Micro Control Manager to check antivirus (AV) and engine status on clients or endpoints using the incident scope.

Integration Capabilities

Action

- Check AV Status on Client
- Check Engine Status
- Check Pattern Status

Configuration

Configuring Trend Micro Control Manager

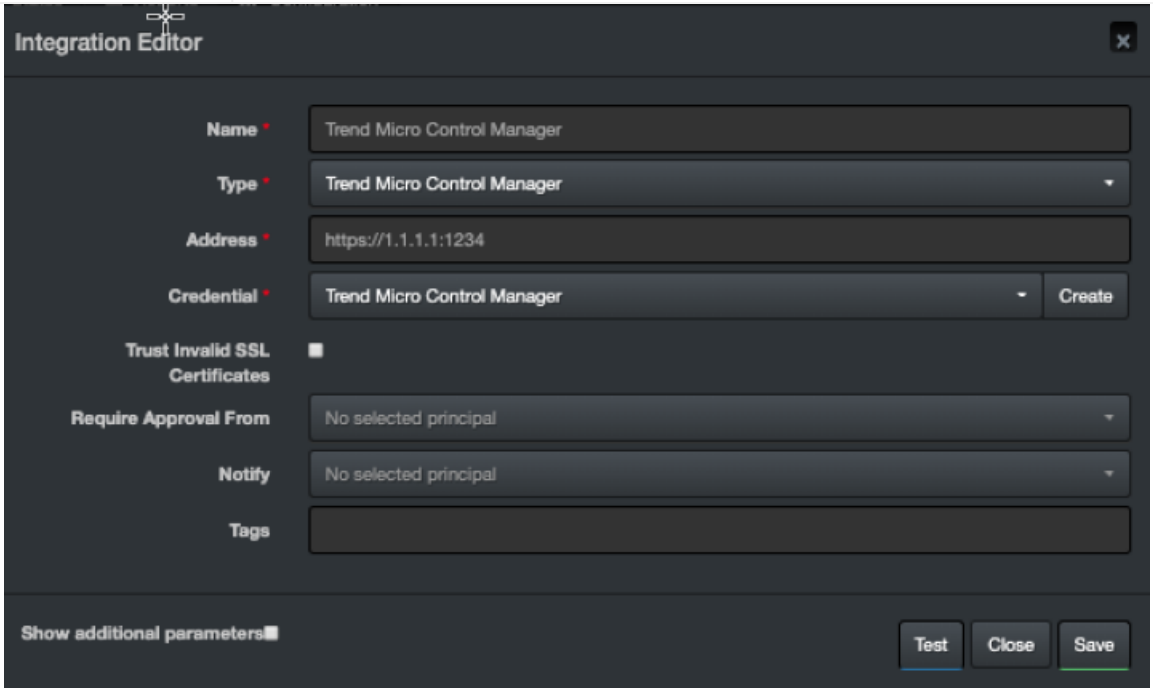
- Access to HTTPS as SOAR connects to the Trend Micro Control Manager's web application using HTTPS
- SOAR user account with admin role
- Queries must be generated on the server used by SOAR

Configuring SOAR

1. Navgate to **Configuration** > **Integrations**.
2. Specify the following parameter values in the **Integrations Editor**:

Parameter	Value
Name	Display name of Trend Micro Control Manager integration on SOAR
Type	Trend Micro Control Manager

Address	Address of the integration (in the following format: http[s]://1.1.1.1:1234[/sdk] or http[s]://abc.example.com:1234[/sdk])
Credential	Name of the credential set created under the **Credentials** menu
Trust Invalid SSL Certificates	Select this if Engine's certificate is self-signed or is not recognized by browsers
Require Approval From	Select users from list who can provide approval before executing actions on this integration
Notify	Select users from the list to notify when SOAR performs an action on this integration



3. Click **Test** to test the integration.
4. Click **Save** to save the integration.

Integration Guide for Turkcell Threat Intelligence or Bozok

Integration Overview

Turkcell Threat Intelligence is a service which lets users to query reputation of Indicators of Compromise such as data leakage, brand protection, and vulnerability modules.

Integration Capabilities

ArcSight SOAR has the following integration capabilities with Turkcell Threat Intelligence:

- Domain Query
- Email Query
- Hash Query
- IP Query

Use Case: Investigating Phishing Campaigns

SOAR integrates with Turkcell Threat Intelligence or Bozok to investigate and mitigate phishing campaigns. SOAR extracts the indicators such as sender address, IP address, and URLs from a phishing report email of the user and creates a new incident on the Incident Management Service Desk. SOAR then checks with Turkcell Threat Intelligence or Bozok if this is a known attack and previously analyzed. This investigation can either be performed automatically within a playbook or manually by an analyst.

Configuration

Prerequisites

- Access to https://bozok.turkcell.com.tr (443/tcp port) as SOAR connects to Turkcell Threat Intelligence/Bozok API through HTTPS
- An API key for SOAR to connect to Turkcell Threat Intelligence/Bozok service

Configuration on Turkcell Threat Intelligence or Bozok

- No specific configuration is needed on Turkcell Threat Intelligence or Bozok.

Configuring SOAR

1. Click **Configuration** > **Credentials** > **Create Credential**.
2. Specify the following parameter values in the **Credential Editor form**:

   a. Internal Credential:

Parameter	Value
Type	Internal Credential
Name	Display name of credential set (For example, Turkcell Threat Intelligence Credentials)
Username	Empty
Password	Empty
Private Key	API key obtained from the service provider

   b. Credential Store:

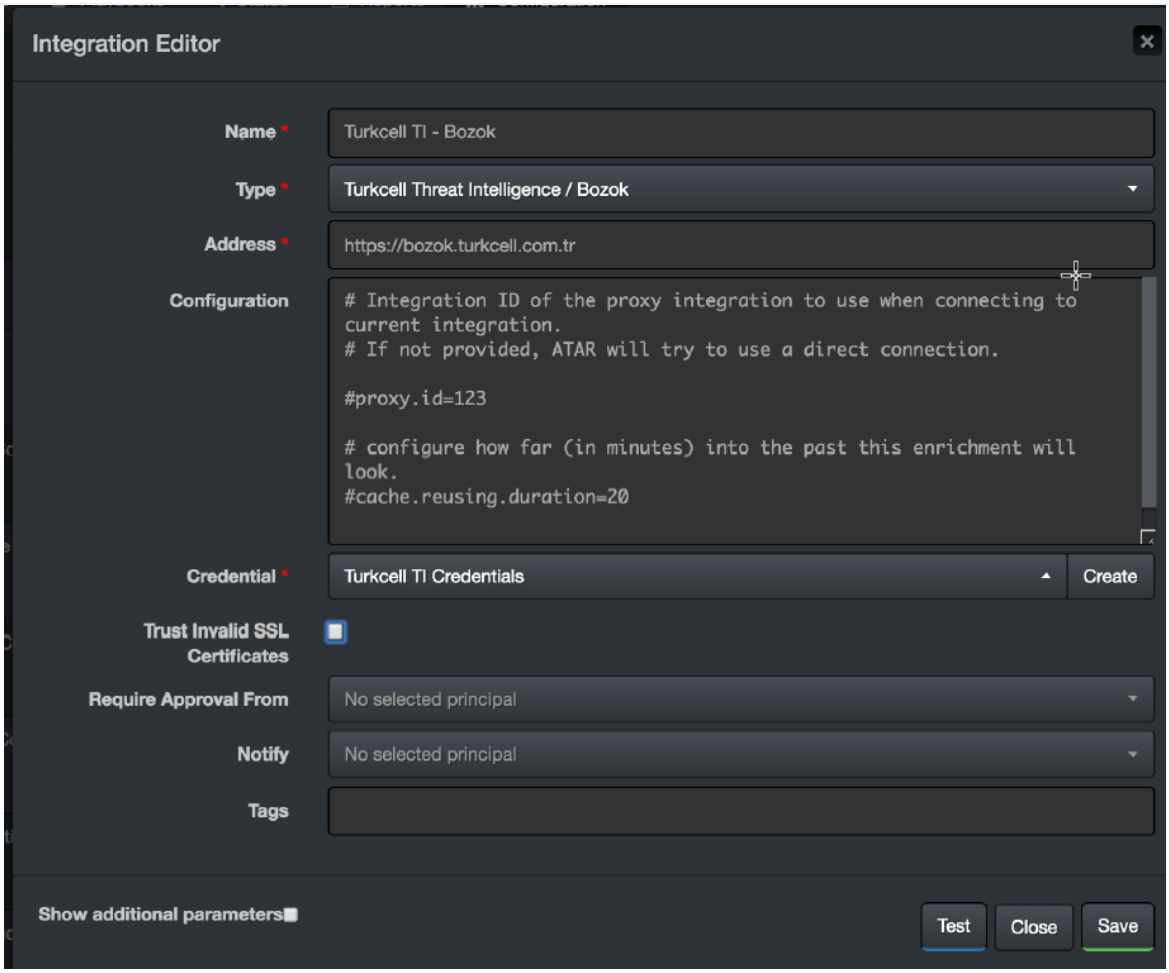
Parameter	Value
Type	External credential
Name	Name of the credential with full path of the safe on store

3. Click **Configuration** > **Integrations** > **Create Integration**.
4. Specify the following parameter values in the **Configuration form**:

Parameter	Value
Name	Display name of Turkcell Threat Intelligence integration on SOAR
Type	Turkcell Threat Intelligence
Address	Address of Turkcell Threat Intelligence service(in the following format: https://bozok.turkcell.com.tr)
Credential	Name of the credential set created (For example, Turkcell Threat Intelligence Credentials)
Trust Invalid SSL Certificates	Unselect

Configuration	Specify the following configuration parameters:
	`# Integration ID of the proxy integration to use`
	`when connecting to current integration.`
	`# If not provided, SOAR will try to use a direct connection.`
	`proxy.id=5434`
	`# configure how far (in minutes) into the past`
	`this enrichment will look.`
	`cache.reusing.duration=60`

Require Approval From	Not applicable as SOAR executes enrichment on Turkcell Threat Intelligence
Notify	Not applicable as SOAR executes enrichment on Turkcell Threat Intelligence



5. Click **Test** to test the integration.
6. Click **Save** to save the integration.

Integration Guide for USOM (TR-CERT) Intelligence Feed

Integration Overview

USOM (TR-CERT- Computer Emergency Response Team of the Republic of Turkey) Intelligence Feed is an actively maintained local feed about various malicious categories prepared by TR-CERT team.

Integration Capabilities

ArcSight SOAR has the following integration capabilities with USOM (TR-CERT):

- Ingest Feed as Alert

Use Case: Blocking malicious URLs and IPs before they do any harm

SOAR integrates with USTA(**need the acronym in this context**) (TR-CERT) intelligence feed to block malicious entities on your perimeter protection before they do any harm.

Configuration

Prerequisites

- Access to https://www.usom.gov.tr/url-list.xml as SOAR connects to USTA (TR-CERT) intelligence feed through HTTPS.

Configuring USOM (TR-CERT) Intelligence Feed

- No specific configuration is needed on USOM (TR-CERT) Intelligence Feed.

Configuring SOAR

Configuring USOM (TR-CERT) Intelligence Feed as Alert Source

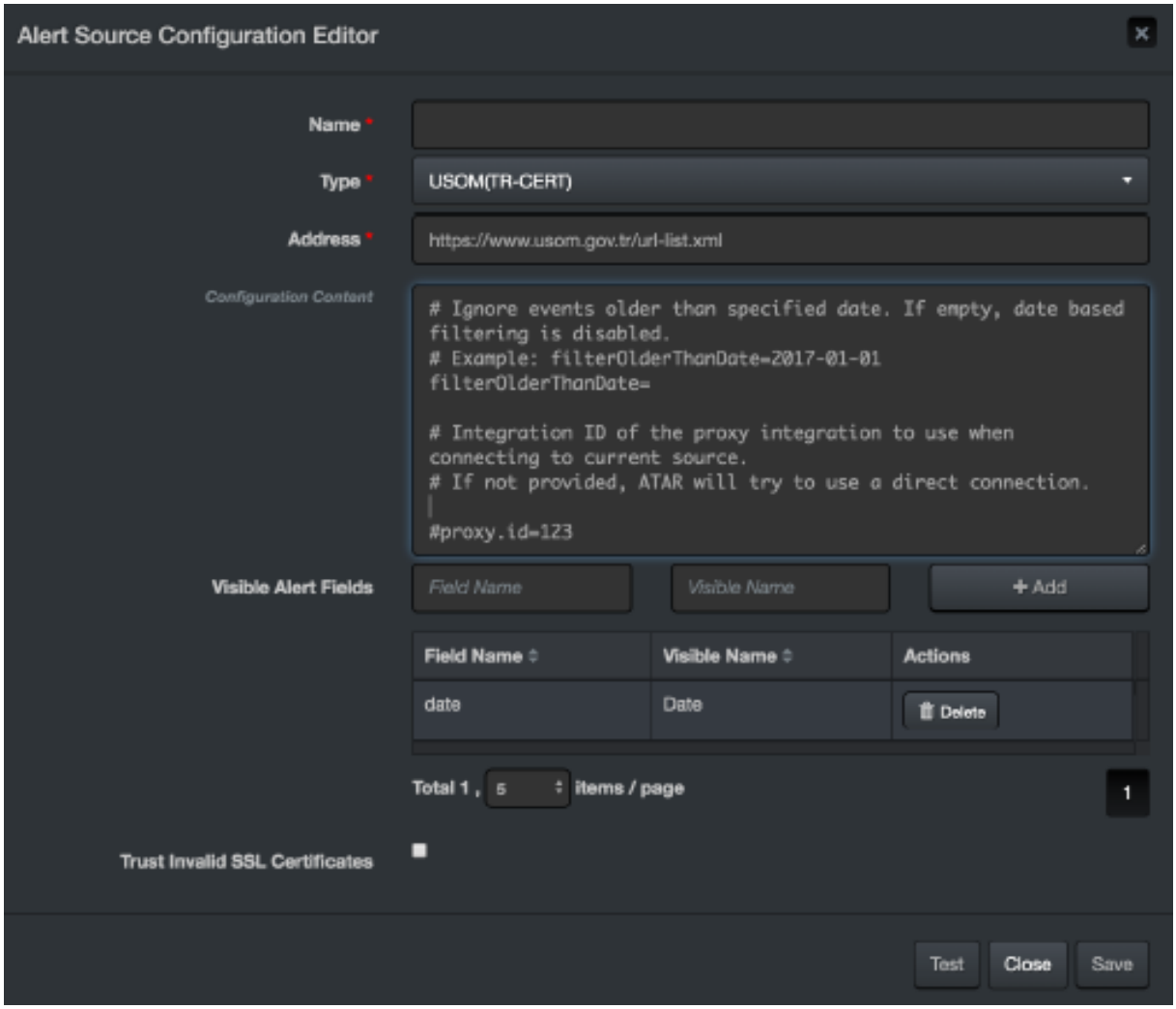
1. Click **Configuration** > **Alert Source**> **Create Alert Source Configuration**.
2. Specify the following parameter values in the **Configuraiton form**:

Parameter	Value
Name	Display name of USOM (TR-CERT) Intelligence Feed Alert Source on SOAR
Type	USOM (TR-CERT)
Address	Address of USOM (TR-CERT) Intelligence Feed (in the following format: https://www.usom.gov.tr/url-list.xml)

Configuration	Specify the following configuration parameters:
	`# Ignore events older than specified date.`
	`If empty, date based filtering is disabled.`
	`# Example: filterOlderThanDate=2017-01-01`
	`filterOlderThanDate=2019-08-01`
	`# Integration ID of the proxy integration to use`
	`when connecting to current source.`
	`# If not provided, SOAR will try to use a direct`
	`connection.`
	`#proxy.id=2443`

Trust Invalid SSL Certificates	Unselect
Visible Alert Fields	Define which alarm fields are displayed Incident Management Service Desk
Notify	Not applicable

3. Click **Test** to test the integration.
4. Click **Save** to save the integration.



Additional Notes

The intelligence feed is specialized for Turkey and is accessible only from IP ranges of Turkey.

Integration Guide for VirusTotal

Integration Overview

VirusTotal inspects suspicious files and URLs to detect types of malware with over seventy antivirus scanners and URLs or domain blacklisting services, in addition to a myriad of tools to extract signals from the studied content.

Integration Capabilities

SOAR has the following integration capability with VirusTotal:

- Domain Query
- Domain/Downloaded Files Query
- Domain/Subdomains Query
- Domain/URLs Query
- File Query
- Hash Query
- IP Query
- IP/Downloaded Files Query
- IP/Passive DNS Query
- IP/URLs Query
- URL Query

Use Case: Blocking access to malicious URL

During the investigation of an attack, SOAR checks for suspicious IP addresses, URLs, files, and hash values to VirusTotal if these indicators are known and previously analyzed. According to returned confidence score, SOAR decides on the next course of action. This investigation can either be performed automatically within a playbook or manually by an analyst.

Configuration

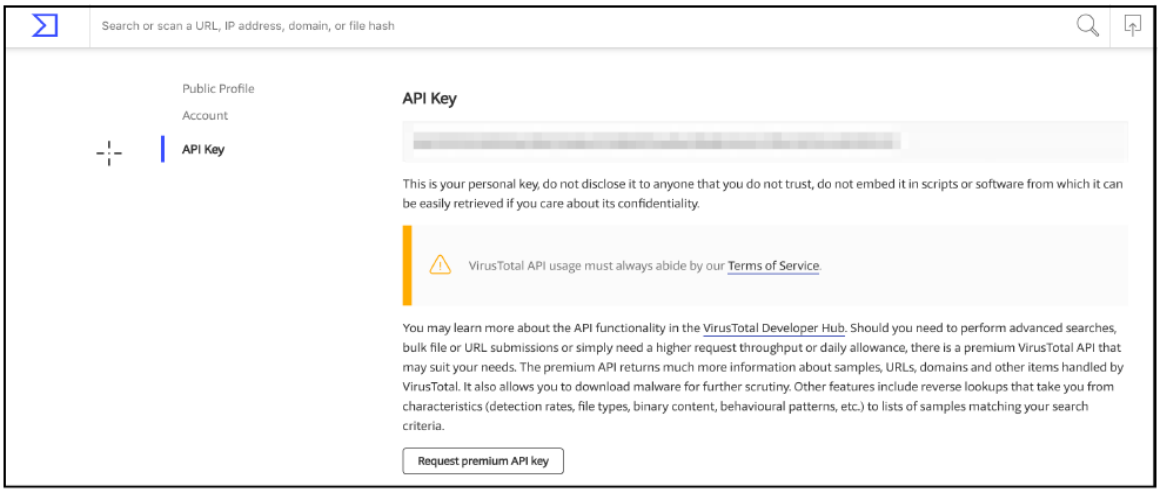
Prerequisites

- VirusTotal API version 3
- Access to tcp port 443 as SOAR connects to VirusTotal API http://www.virustotal.com

- An API key for SOAR to connect to VirusTotal

Configuring VirusTotal

- No specific configuration is needed on VirusTotal.
- Login to https://www.virustotal.com with your username and make a note of the API key under **Settings**> **API Key**.



Configuring SOAR

1. Click **Configuration** > **Credentials** > **Create Credential**.
2. Specify the following parameter values in the **Credential Editor**:
   a. **Internal Credential**

Parameter	Value
Type	Internal Credential
Name	Display name of credential set (For example, VirusTotal Credentials)
Username	Empty
Password	Empty
Private Key	API Key you have on VirusTotal

   b. **Credential Store:**

Parameter	Value
Type	External Credential
Name	Name of the credential with pull path of the safe on store

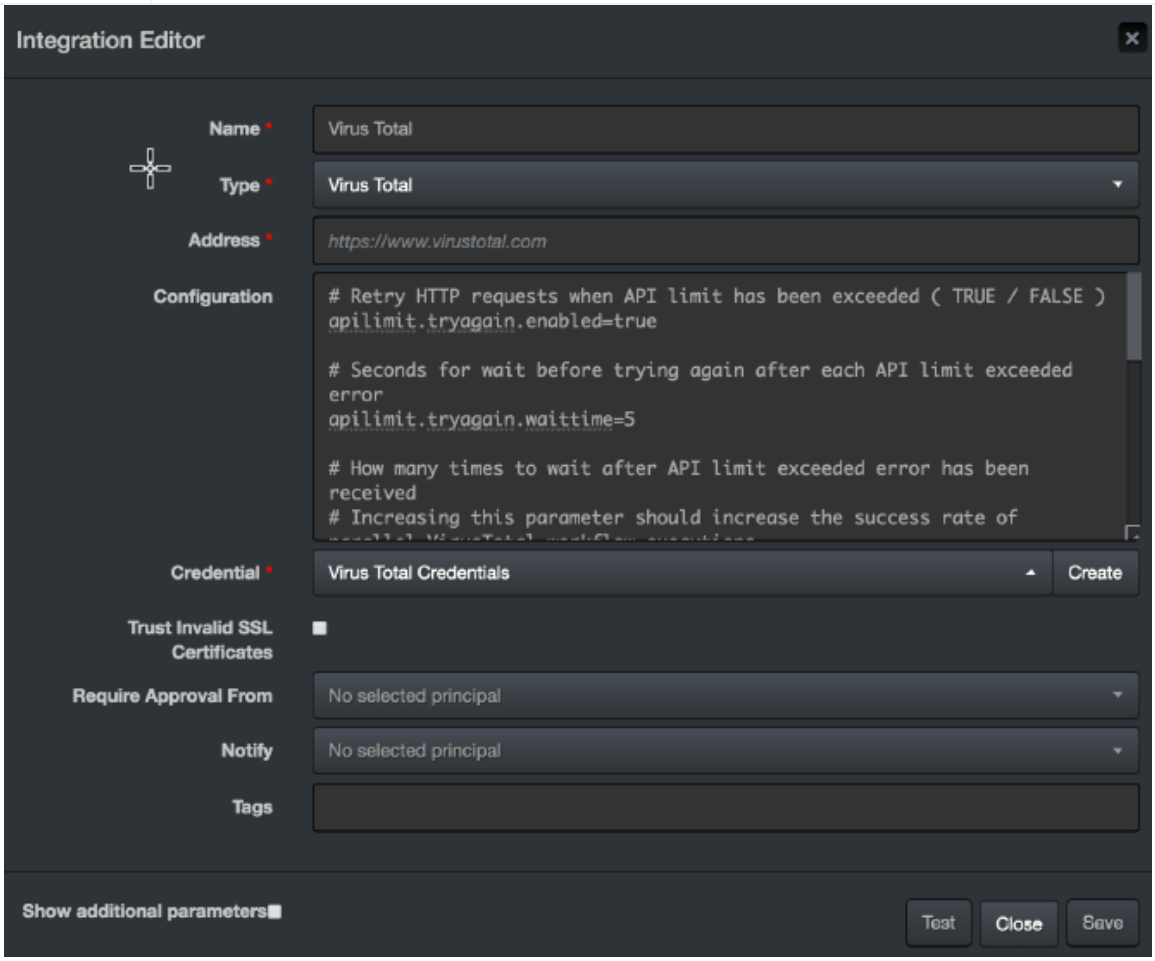
3. Navigate to **Configuration** > **Integrations** > **Create Integration**.

4. Specify the following parameter values in the **Configuration form**:

Parameter	Value
Name	Display name of VirusTotal integration on SOAR
Type	VirusTotal
Address	Address of the integration (in the following format https://www.virustotal.com)

Configuration	Specify the following configuration parameters:
	`# Retry HTTP requests when API limit has`
	`been exceeded ( TRUE / FALSE )`
	`apilimit.tryagain.enabled=true`
	`# Seconds for wait before trying again`
	`after each API limit exceeded error`
	`apilimit.tryagain.waittime=5`
	`apilimit.tryagain.waittime=5`
	`# How many times to wait after API limit`
	`exceeded error has been received`
	`# Increasing this parameter should increase`
	`the success rate of parallel VirusTotal workflow`
	`apilimit.tryagain.waitlimit=3`
	`# Integration ID of the proxy integration`
	`to use when connecting to current integration.`
	`# If not provided, ATAR will try to use a`
	`direct connection.`
	`#proxy.id=123`
	`# configure how far (in minutes) into`
	`the past this enrichment will look.`
	`#cache.reusing.duration=20`
	`# Enrichment timeout duration after`
	`start time ( in seconds )`
	`scan.query.timeout=3600`
	`# Expiration period of hash scans`
	`# If not provided, ATAR will use 30 days`
	`by default`
	`#scan.result.expiration.period.in.days=30`
	`# VirusTotal APIv3 parameter`
	`# Limits page count for relation queries.`
	`SOAR will use 1 page by default`
	`#scan.result.page.count.max=1`

Trust Invalid SSL Certificates	Unselect
Require Approval From	Not applicable
Notify	Not applicable



5. Click **Test** to test the integration.
6. Click **Save** to complete the integration.

Additional Notes

- Domain and IP-related queries retrieve results in 40-item batches. For some IOCs, this may result in too many consecutive queries and long query-times.
- The file queries are limited to 32MB due to limits with VirusTotal API.
- Domain or URLs, Domain or Downloaded Files, IPor URLs, and IP or Downloaded Files only return the scope items with confidence score greater than 0.

Integration Guide for VMware ESXi

Integration Overview

SOAR uses VMware ESXi(Elastic Sky X integration) to perform some actions on the virtual machines (VMs).

Integration Capabilities

Action

- Create Snapshot of a VM
- Export VM
- Get Information of All VMs
- Power On VM
- Power Off VM
- Reset VM
- Reboot VM
- Standby VM
- Suspend VM

Configuration

Configuring VMware ESXi

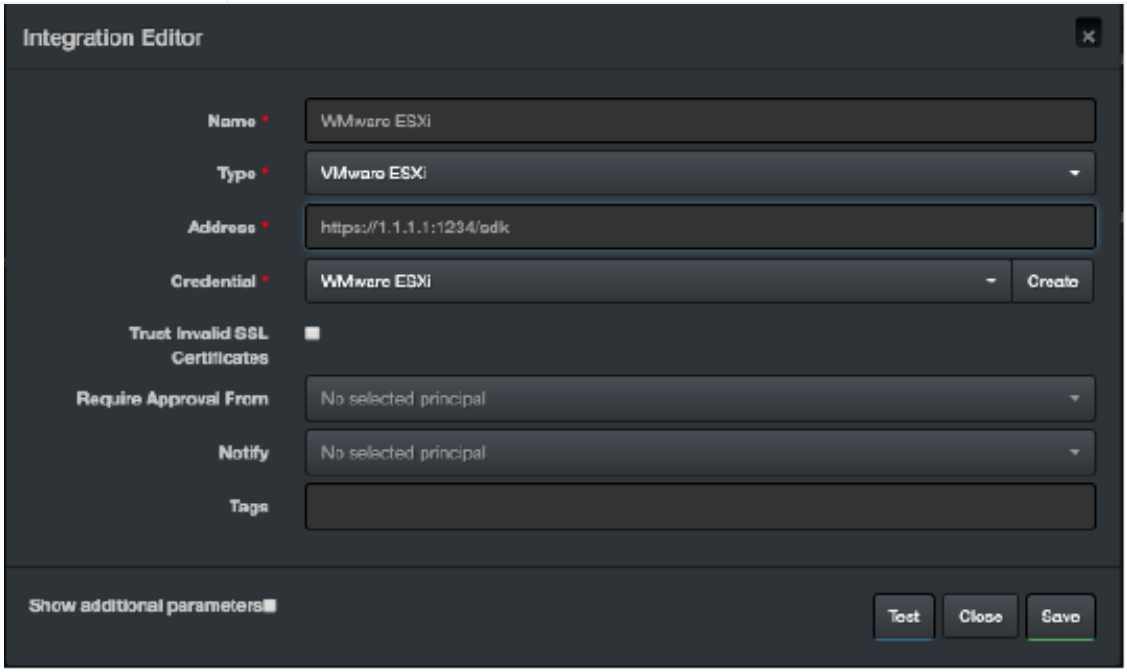
- Access to HTTPs for SOAR to connect to VMware ESXi Server's SDK
- SOAR account with admin role

Configuring SOAR

1. Navigate to **Configuration** > **Integrations**.
2. In the Integrations Editor, specify the following parameter values:

Parameter	Value
Name	Display name of VMware ESXi integration on SOAR

Type	VMware ESXi
Address	Address of the integration (in the following format: http[s]://1.1.1.1:1234[/sdk] or http[s]://abc.example.com:1234[/sdk])
Credential	Credential defined for the integration under the Credentials menu
Trust Invalid SSL Certificates	Select this if Engine's certificate is self-signed or is not recognized by browsers
Require Approval From	Select users from the list who can provide approval before executing actions on this integration
Notify	Select users from the list to notify when SOAR performs an action on this integration



3. Click **Test** to test the integration.
4. Click **Save** to complete the integration.

Integration Guide for VxStream Sandbox

Integration Overview

VxStream Sandbox is an automated malware analysis system that includes the unique Hybrid Analysis technology. It is available as a standalone software package that is automatically deployed within your local infrastructure and operates without an external dependency or callback mechanism. It is possible to execute files on any Windows guest image (For example, a copy of your local workstation) and has a variety of integration and interface capabilities.

The feature set of VxStream Sandbox is extensive, with hundreds of generic indicators at its core. It detects unknown threats independent of Anti-Virus signatures. Empowered by Hybrid Analysis, the entire process memory gets analyzed using multiple timed snapshots, including the runtime sample. This feature allows the extraction of more indicators (Strings/API calls) regardless of execution. This approach enables the analysis of dormant code, evasive conditions, and extracts more valuable IOCs.

Integration Capabilities

Action

- Hash analysis

Configuration

Configuration on VxStream Sandbox

- Access to HTTPs for SOAR to connect to VxStream Sandbox

Configuring SOAR

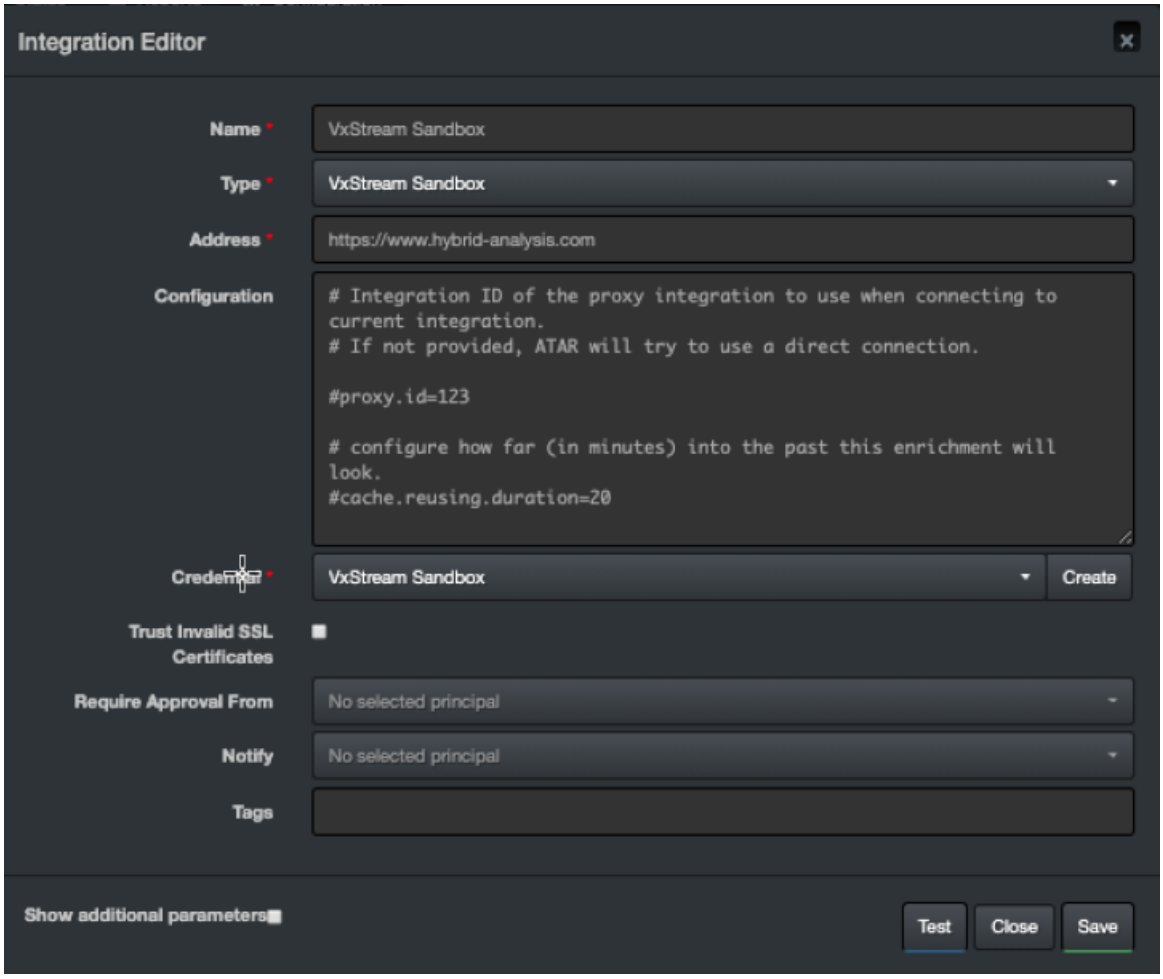
1. Navigate to **Configuration** > **Integrations**.
2. In the **Integrations Editor** window, specify the following parameter values:

Parameter	Value
Name	Display name of VxStream Sandbox integration on SOAR
Type	VxStream Sandbox

Address	Address of the integration (in the following format: https://www.hybrid-analysis.com)

Configuration	Specify the following configuration parameters:
	`# Integration ID of the proxy integration to use when connecting to`
	`# current integration.`
	`# If not provided, ATAR will try to use a direct connection.`
	`#proxy.id=123`
	`# configure how far (in minutes) into the past this enrichment will look.`
	`#cache.reusing.duration=20`

Credential	Credential defined for the integration under the Credentials menu
Trust Invalid SSL Certificates	Select this if Engine's certificate is self-signed or is not recognized by browsers
Require Approval From	Select users from the list who can provide approval before executing actions on this integration
Notify	Select users from the list to notify when SOAR performs an action on this integration



3. Click **Test** to test the integration.
4. Click **Save** to complete the integration.

Integration Guide for WinRM

Integration Overview

This appendix provides a detailed, step-by-step configuration procedure to enable SOAR to properly work with WinRM.

Configuration On Domain-Controller

- **To create a Group Policy object for your domain:**

1. Navigate to **Start** > **Control Panel**.
2. In the Control Panel, select **Administrative Tools** > **Group Policy Management**.
3. From the menu tree, click **Domains** > **[your domain's name]**.
4. Right-click and select **Create a GPO in this domain, and Link it here**.
5. Input **WinRM-SOAR**.
6. Execute the following command:

```
reg add HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System /v
LocalAccountTokenFilterPolicy /t REG_DWORD /d 1 /f
```

7. Click **OK**.

- **To edit the new Group Policy object you created:**

1. Right-click on the new WinRM-SOAR Group Policy Object and select **Edit**.
2. From the menu tree, click **Computer Configuration** > **Policies**.
3. In the **Policies**, click **Administrative Templates: Policy definitions** > **Windows Components** > **Windows Remote Management (WinRM).**
4. Navigate to **WinRM Service**.

Note: After editing the Group Policy Object, three WinRM service settings are enabled.

   a. **To Allow remote server management through WinRM**

      i. Right-click either **Allow automatic configuration of listeners(Windows Server 2010)** or **Allow remote server management through WinRM(Windows Server 2012)**

ii. Click **Edit**.
iii. To allow remote server management through WinRM, select **Enabled**.
iv. Enter an asterisk (*) in each field.
v. Click OK.

b. **To Allow unencrypted traffic through WinRM**
i. Right-click **Allow unecrypted traffic** and click **Edit**.
ii. Select **Enabled** and click **OK**.

Now the Windows Remote Management is enabled on the Group Policy.

c. **To Enable the Service that goes with it**
i. In the **Group Policy Management Editor window**, click **Preferences** > **Control Panel Settings** > **Services**.
ii. Right-click **Services and select New** > **Service**.
iii. Select **Automatic** as the startup.
iv. Enter **WinRM** as the service name.
v. Select **Start service** as the service action.
vi. Select **This account** to log in as.
vii. Enter **NT AUTHORITY\NetworkService** as the user and use **a space character** as the password.
viii. Click **OK**.

- **To allow inbound remote administration by updating the firewall rules**:

The steps enable the following firewall rules:

- Windows Firewall: Allow inbound remote administration exception
- Windows Firewall: Allow ICMP exception

1. In the **Group Policy Management Editor**, click **Computer Configuration** > **Policies**.
2. Click **Administrative Templates: Policy definitions** > **Network** > **Network Connections** > **Windows Firewall** > **Domain Profile**.
3. Right-click **Windows Firewall: Allow inbound remote administration exception** and click **Edit**.
4. Select **Enabled**.
5. Enter an asterisk (*) into each field and click **Ok**.

6. Right-click **Windows Firewall: Allow ICMP exception** and click **Edit**.
7. Select **Enabled**.
8. Select **Allow inbound echo request** and click **Ok**.

- **To create a new inbound firewall rule and update the network list manager for unidentified networks:**

1. Click **Computer Configuration** > **Windows Settings** > **Security Settings** > **Windows Firewall with Advanced Security** > **Inbound Rules.**
2. Right-click **Inbound Rules** and click **New Rule**.
3. Select **Predefined**.
4. Select **Windows Remote Management** from the list of services.
5. Click **Next**.
6. Unselect the entry profile **Public** and click **Next.**
7. Click **Finish**.
8. Right-click the new rule and click **Properties**.
9. Click the **Advanced** tab and unselect all and select **Private**.
10. Click the **Scope tab.**
11. Check these IP addresses on Remote IP Address and specify IP address of the SOAR machine and click **OK**.
12. From the menu tree, click **Computer Configuration** > **Windows Settings** > **Security Settings** > **Network List Manager Policies**.
13. Right-click **Unidentified Networks** and click **Properties.**
14. Select the **Location type** to **Private** and click **Ok**.

Configuring SOAR

Use the format *username|Computer name* as WinRM credentials. For example, *localadmin|DEV-EXCHANGE18*.

Configuring Domain-Controller for WinRM HTTPS Transport

1. Open the Certificate Authority management console.
2. Right-click **Certificate Templates** and select **Manage**.
3. In the template management console, scroll down and select **Web Server template**.
4. Right-click **Web Server Template**, select **Duplicate Template**.
5. In the **Certificate Property Window** for the new template, navigate to the **General Tab**.

6. Set **Display Name** and **Template Name** to **SOARWINRMHTTPS**.

   Note: Use the same name without spaces. If there is a space that leads to a bug where the process to enroll a new certificate repeats.
7. In the **Subject Name** tab, select **Build from this Active Directory information**.
8. In the **Subject name format** select **Common Name** and select **DNS name**.
9. Click **Security** > specify the **Domain Computers** group for the domain. Allow Read, Enroll and Autoenroll and click **OK**.
10. In the **Certificate Authority management console**, right-click **Certificate Templates** and select **New Template**.
11. Double-click **SOARWINRMHTTPS** and close the window.
12. Navigate to **Start** > **Control Panel**.
13. Select **Administrative Tools** and **Group Policy Management**.
14. In the Menu tree, click **Domains** > **[your domain's name]**.
15. Create a batch script for starting WinRM HTTPS Listener named **SoarWinRMSSLStartupScript.ps1**.
16. Copy and paste the following code into **AtarWinRMSSLStartupScript.ps1**:

```
Start-Transcript C:\Scripts\transaction.log

$sysinfo = Get-WmiObject -Class Win32_ComputerSystem

$server = "{0}.{1}" -f $sysinfo.Name, $sysinfo.Domain

$LatestThumb = Invoke-Command -ScriptBlock {

Get-ChildItem -Path Cert:\LocalMachine\My |

where {$_.subject -match "CN=$server"}

Sort-Object -Property NotAfter -Descending |

Select-Object -Last 1 -ExpandProperty Thumbprint

} -ErrorAction Stop

#If HTTPS Listener does not exist create Listener with quick config.Else
evaluate

# available certificates ,sort them by expire date , select first
thumbprint

$result=(((Get-ChildItem -Path WSMan:\localhost\Listener).keys) -match
'HTTPS')

if($result.Count -eq 0) {
```

```
Set-WSManQuickConfig -UseSSL -Force
} else {
Set-WSManInstance -ResourceURI winrm/config/Listener \
-SelectorSet @{Address="*";Transport="HTTPS"} \
-ValueSet @{CertificateThumbprint=$LatestThumb.Thumbprint[1]}
Restart-Service -Force -Name WinRM
}
Stop-Transcript
```

17. Navigate to **Start** > **Control Panel**.
18. Select Administrative Tools > Group Policy Management.
19. Right-click **WinRM-SOAR** and click **Edit**.
20. Click **Computer Configuration** > **Policies** > **Windows Settings** > **Security Settings** > **Public Key Policies**.
21. Double-click **Certificate Services Client - Auto-Enrollment**.
22. Set the **Configuration Model** to **Enable**.
23. Select **Renew expired certificates**, **update pending certificates**, and **remove revoked**certificates.and **Update certificates that use certificate templates**.
24. Click **Ok.**
25. Click **Computer Configuration** > **Policies** > **Windows Settings** > **Scripts.**
26. Double-click **Startup**.
27. In the **PowerShell Scripts**, click **Add**> **Browse** the file **named AtarWinRMSSLStartupScript.ps1**. and click **OK**.

Force Group Policy Update

Use the following PowerShell commands to force a Policy Update as described in the command block:

```
$computers = Get-ADComputer -Filter *
$computers | ForEach-Object -Process {Invoke-GPUpdate -Computer $_.name \
-RandomDelayInMinutes 0 -Force}
```

Additional Notes

The following patch must be applied to the target computer for WinRM to work without an error:

https://support.microsoft.com/en-us/kb/2842230

Support

Contact Information

Phone	A list of phone numbers is available on the Technical Support Page: https://softwaresupport.softwaregrp.com/support-contact-information
Support Web Site	https://softwaresupport.softwaregrp.com/
ArcSight Product Documentation	https://community.softwaregrp.com/t5/ArcSight-Product-Documentation/ct-p/productdocs

Send Documentation Feedback

If you have comments about this document, you can contact the documentation team by email. If an email client is configured on this computer, click the link above and an email window opens with the following information in the subject line:

Feedback on Integration Guides (SOAR 3.1 3.1)

Just add your feedback to the email and click send.

If no email client is available, copy the information above to a new message in a web mail client, and send your feedback to Documentation-Feedback@microfocus.com.

We appreciate your feedback!