



Protegrity Data Security Gateway 3.1.0.5

Created on: Nov 19, 2024

Notice

Copyright

Copyright © 2004-2024 Protegrity Corporation. All rights reserved.

Protegrity products are protected by and subject to patent protections;

Patent: <https://www.protegrity.com/patents>.

The Protegrity logo is the trademark of Protegrity Corporation.

NOTICE TO ALL PERSONS RECEIVING THIS DOCUMENT

Some of the product names mentioned herein are used for identification purposes only and may be trademarks and/or registered trademarks of their respective owners.

Windows, Azure, MS-SQL Server, Internet Explorer and Internet Explorer logo, Active Directory, and Hyper-V are registered trademarks of Microsoft Corporation in the United States and/or other countries.

Linux is a registered trademark of Linus Torvalds in the United States and other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

SCO and SCO UnixWare are registered trademarks of The SCO Group.

Sun, Oracle, Java, and Solaris are the registered trademarks of Oracle Corporation and/or its affiliates in the United States and other countries.

Teradata and the Teradata logo are the trademarks or registered trademarks of Teradata Corporation or its affiliates in the United States and other countries.

Hadoop or Apache Hadoop, Hadoop elephant logo, Hive, and Pig are trademarks of Apache Software Foundation.

Cloudera and the Cloudera logo are trademarks of Cloudera and its suppliers or licensors.

Hortonworks and the Hortonworks logo are the trademarks of Hortonworks, Inc. in the United States and other countries.

Greenplum Database is the registered trademark of VMware Corporation in the U.S. and other countries.

Pivotal HD is the registered trademark of Pivotal, Inc. in the U.S. and other countries.

PostgreSQL or Postgres is the copyright of The PostgreSQL Global Development Group and The Regents of the University of California.

AIX, DB2, IBM and the IBM logo, and z/OS are registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide.

Utimaco Safeware AG is a member of the Sophos Group.

Xen, XenServer, and Xen Source are trademarks or registered trademarks of Citrix Systems, Inc. and/or one or more of its subsidiaries, and may be registered in the United States Patent and Trademark Office and in other countries.

VMware, the VMware “boxes” logo and design, Virtual SMP and VMotion are registered trademarks or trademarks of VMware, Inc. in the United States and/or other jurisdictions.

Amazon Web Services (AWS) and AWS Marks are the registered trademarks of Amazon.com, Inc. in the United States and other countries.

HP is a registered trademark of the Hewlett-Packard Company.

HPE Ezmeral Data Fabric is the trademark of Hewlett Packard Enterprise in the United States and other countries.

Dell is a registered trademark of Dell Inc.

Novell is a registered trademark of Novell, Inc. in the United States and other countries.

POSIX is a registered trademark of the Institute of Electrical and Electronics Engineers, Inc.

Mozilla and Firefox are registered trademarks of Mozilla foundation.

Chrome and Google Cloud Platform (GCP) are registered trademarks of Google Inc.

Swagger Specification and all public tools under the swagger-api GitHub account are trademarks of Apache Software Foundation and licensed under the Apache 2.0 License.

Table of Contents

Copyright.....	2
Chapter 1 Introduction to this Guide.....	10
1.1 Sections contained in this Guide.....	10
1.2 Accessing the Protegility documentation suite.....	11
1.2.1 Viewing product documentation.....	11
1.2.2 Downloading product documentation.....	12
Chapter 2 Protegility Gateway Technology.....	13
2.1 Background.....	13
2.2 Audience.....	13
2.3 What is the Protegility Gateway Technology?.....	14
2.4 How do the Protegility Gateways Protect Data?.....	15
2.5 Why the Protegility Gateway Technology?.....	16
2.6 Data-Centric Auditing & Protection Platform.....	17
2.7 Configuration over Programming (CoP) brings it all together.....	18
Chapter 3 Protegility Gateway Technology Products.....	20
3.1 Cloud Security Gateway.....	20
3.2 Data Security Gateway.....	22
3.2.1 Data Security Gateway Scenarios.....	22
3.2.1.1 Data Security Gateway used to protect Web Applications.....	22
3.2.1.2 Data Security Gateway as an API Security Gateway.....	23
3.2.1.3 Data Security Gateway for Files.....	23
3.2.1.4 Data Security Gateway as an On Demand RESTful API Server.....	24
Chapter 4 Technical Architecture.....	25
4.1 System Architecture.....	25
4.1.1 Platform Layer.....	26
4.1.2 Data Collection Layer.....	26
4.1.3 Data Extraction Layer.....	27
4.1.4 Action Layer.....	27
4.2 Configuration over Programming (CoP) Architecture.....	27
4.2.1 CoP Overview.....	27
4.2.2 CoP Ruleset.....	29
4.2.3 Ruleset Structure.....	31
4.2.4 Ruleset Tree of Trees (ToT).....	31
4.2.5 Dynamic Configuration over Programming (CoP).....	32
4.2.5.1 Dynamic CoP structure.....	34
4.2.5.2 Dynamic CoP Usecase.....	35
4.2.6 Ruleset Execution Engine.....	40
4.2.7 Ruleset and Ruleset Execution Example.....	41
4.3 Codebook Re-shuffling.....	41
4.3.1 Codebook Re-shuffling in the PEP Server.....	44
4.3.2 Re-protecting the Existing BLOB.....	46
Chapter 5 Deployment Scenarios.....	48
Chapter 6 Protegility Methodology.....	50
6.1 Data Governance.....	50
6.2 Discovery.....	50

6.3 Solution Design.....	51
6.4 Product Installation.....	51
6.5 Solution Configuration.....	51
6.6 Initial Migration.....	51
6.7 Testing.....	51
6.8 Production Rollout.....	52
Chapter 7 Planning for Gateway Installation.....	53
7.1 Planning Overview.....	53
7.2 Minimum Hardware Requirements.....	53
7.3 ESA.....	54
7.4 Forwarding Logs in DSG.....	54
7.5 LDAP and SSO Configurations.....	55
7.5.1 Enabling SSO on DSG.....	55
7.5.2 Configuring SPNEGO Authentication on the Web Browser.....	56
7.5.2.1 Configuring SPNEGO Authentication on Firefox.....	56
7.5.2.2 Configuring SPNEGO Authentication on Internet Explorer.....	57
7.5.2.3 Configuring SPNEGO Authentication on Chrome.....	57
7.5.3 Logging to the Appliance.....	57
7.6 Mapping of Sensitive Data Primitives.....	58
7.7 Network Planning.....	58
7.7.1 NIC Bonding.....	63
7.7.1.1 Bonding Modes.....	63
7.7.1.2 Prerequisites.....	63
7.7.1.3 Creating a Bond.....	64
7.7.1.4 Removing a Bond.....	65
7.7.1.5 Viewing a Bond.....	66
7.7.1.6 Resetting the Bond.....	67
7.8 HTTP URL Rewriting.....	67
7.9 Clustering and Load Balancing.....	69
7.10 SSL Certificates.....	70
Chapter 8 Installing the DSG.....	71
8.1 Installing the DSG On-Premise.....	71
8.1.1 Installing DSG.....	72
8.2 Installing the DSG on Cloud Platforms.....	85
8.2.1 Installing Data Security Gateway (DSG) on Amazon Web Services (AWS).....	85
8.2.1.1 Prerequisites.....	86
8.2.1.2 Backing Up and Restoring the DSG Instance Snapshot on AWS.....	87
8.2.1.3 Installing and Launching DSG.....	87
8.2.1.4 Best Practices for Using DSG on AWS.....	92
8.2.2 Installing the Data Security Gateway (DSG) on Microsoft Azure.....	93
8.2.2.1 Prerequisites.....	93
8.2.2.2 Backing Up and Restoring the DSG Instance Snapshot on Azure.....	94
8.2.2.3 Installing and Launching DSG.....	95
8.2.2.4 Azure Cloud Utility.....	104
8.2.3 Installing the Data Security Gateway (DSG) on Google Cloud Platform (GCP).....	105
8.2.3.1 Prerequisites.....	105
8.2.3.2 Backing Up and Restoring the DSG Instance Snapshot on GCP.....	107
8.2.3.3 Installing and Launching DSG.....	107
8.2.4 Installing the Cloud Utility AWS Tool.....	112
8.2.5 Post DSG installation Steps.....	113
8.2.5.1 Verifying the DSG installation.....	114
8.2.5.2 Pushing the DSG Rulesets.....	114
8.2.5.3 Verifying the Startup Logs.....	114
8.3 Extending ESA with DSG Web UI.....	115
8.4 Setting up ESA Communication.....	115



8.5 Configuring Default Gateway for Network Interfaces.....	118
8.5.1 Configuring Default Gateway for Management NIC (ethMNG) using the DSG CLI Manager.....	118
8.5.2 Configuring Default Gateway for Service NIC (ethSRV0) using the DSG CLI Manager.....	118
8.6 Configuring the DSG Cluster.....	118
8.6.1 Adding a Node to the Cluster.....	121
8.6.2 Applying a DSG Patch.....	123
8.6.3 Applying an Appliance Framework Patch.....	124
8.6.4 Creating a Cluster.....	124
8.6.5 Removing a Node from the Cluster.....	127
8.7 Forwarding Logs to the Audit Store.....	127
8.7.1 Forwarding Appliance Logs to the Audit Store.....	128
8.7.2 Forwarding Audit Logs to the Audit Store.....	130
8.8 Advanced Settings.....	132
8.8.1 Forking DSG Processes.....	133
Chapter 9 Upgrading to DSG v3.1.0.5.....	134
9.1 Backing Up the DSG Appliance OS from the Web UI.....	136
9.2 Upgrading to DSG v3.1.0.5 from DSG v3.1.0.4.....	137
9.3 Upgrading to DSG v3.1.0.5 from DSG v3.1.0.3.....	139
9.4 Upgrading to DSG v3.1.0.5 from DSG v3.1.0.2.....	142
9.5 Upgrading to DSG v3.1.0.5 from DSG v3.1.0.1.....	144
9.6 Upgrading to DSG v3.1.0.5 from DSG v3.1.0.0.....	147
9.7 Upgrading to DSG v3.1.0.5 from DSG v3.0.0.0.....	150
9.8 Upgrading to DSG v3.1.0.5 from DSG v2.6.0.1.....	153
9.9 Upgrading to DSG v3.1.0.5 from DSG v2.6.0.0.....	155
9.10 Upgrading to DSG v3.1.0.5 from DSG v2.4.2.....	158
9.11 Upgrading to DSG v3.1.0.5 from DSG v2.4.1.....	161
9.12 Upgrading to DSG v3.1.0.5 from DSG v2.4.0.....	164
9.13 Restoring the DSG Appliance OS Backup.....	166
9.14 Pushing Audit Logs to Forensics.....	167
9.15 Managing the PEP Server Configuration File.....	167
9.16 Restarting the DSG Node.....	168
9.17 Restoring LogForwarder Custom Files.....	168
9.18 Restoring the Backed Up Codebook Reshuffling Configuration files.....	168
9.19 Verifying UDF Rules for Blocked Modules and Methods.....	170
9.20 Products Compatibility Matrix.....	171
Chapter 10 DSG Web UI.....	172
10.1 Introducing the Cloud Gateway Menu.....	172
10.1.1 Cluster Menu.....	173
10.1.1.1 Monitoring tab.....	173
10.1.1.2 Log Viewer.....	179
10.1.2 Ruleset Menu.....	182
10.1.2.1 RuleSet tab.....	182
10.1.2.2 Learn Mode.....	191
10.1.3 Transport Menu.....	194
10.1.3.1 Certificates/Key Material tab.....	194
10.1.3.2 Tunnels tab.....	200
10.1.4 Global Settings.....	232
10.1.4.1 Debug tab.....	233
10.1.4.2 Global Protocol Stack tab.....	236
10.1.4.3 Web UI tab.....	237
10.1.5 Tokenization Portal.....	238
10.1.6 Additional Configurations using the gateway.json file.....	240
Chapter 11 Overview of Sub-Clustering.....	247

11.1 Sub-Clustering FAQs.....	249
Chapter 12 Implementation.....	251
12.1 Network Setup.....	251
12.1.1 Domain Name System.....	251
12.1.2 Network Connectivity.....	252
12.1.3 SSL Certificates.....	252
12.2 Configuring Ruleset.....	253
12.2.1 Creating a Service under RuleSet.....	253
12.2.1.1 Trusting Self-Signed Certificates for an Outbound Communication.....	254
12.2.2 Use Learn Mode to Find Message Carrying Sensitive Data.....	255
12.2.2.1 Password Masking.....	256
12.2.2.2 Protecting Sensitive Data.....	260
12.3 Forwarding logs to SIEM systems.....	267
12.3.1 Forwarding logs to an external SIEM.....	267
12.3.2 Forwarding logs to AWS CloudWatch.....	268
12.4 Rolling Out to Production.....	268
Chapter 13 Transaction Metrics Logging.....	270
13.1 Total Time Breakdown for HTTP Request.....	276
Chapter 14 Error Metrics Logging.....	278
14.1 Error Metrics with Non-Permissive Errors.....	281
14.2 Configuring the HTTP Status Codes.....	282
Chapter 15 Best Practices.....	283
15.1 Naming Convention.....	283
15.2 Learn Mode.....	283
15.3 Using Prefix and Suffix.....	284
15.4 Profile Reference.....	284
15.5 Modifying Out-of-the-box Profiles and Rules.....	284
15.6 Defining Services.....	284
15.7 Default Certificates and Keys.....	285
15.8 Migration of Data Tokenized by DSG.....	286
Chapter 16 RuleSet Reference.....	287
16.1 Services.....	287
16.1.1 Gateway Service Fields.....	287
16.1.2 REST API.....	289
16.1.3 HTTP Gateway.....	293
16.1.4 Secure WebSocket (WSS) Service.....	298
16.1.5 SMTP Gateway.....	300
16.1.6 SFTP Gateway.....	301
16.1.7 Amazon S3 Gateway.....	305
16.1.8 Mount File System Out-of-Band Service.....	307
16.2 Profiles.....	310
16.3 Rules.....	311
16.3.1 Actions.....	311
16.3.1.1 Exit.....	312
16.3.1.2 Extract.....	312
16.3.1.3 Log.....	347
16.3.1.4 Profile Reference.....	350
16.3.1.5 Set User Identity.....	351
16.3.1.6 Set Context Variable.....	351
16.3.1.7 Dynamic Injection.....	358

16.3.1.8 Error.....	361
16.3.1.9 Transform.....	362
Chapter 17 DSG REST API.....	376
17.1 Overview.....	376
17.2 REST API Authentication.....	377
17.2.1 Basic Authentication.....	377
17.2.1.1 Enabling Rest API Authentication.....	379
17.2.1.2 Enabling a Client Certificate for a Rule.....	380
17.2.2 TLS Mutual Authentication.....	381
17.2.2.1 Enabling Mutual Authentication.....	382
17.3 Protecting an XML Document through DSG REST API.....	382
17.4 Java Client using Native java.net.HttpURLConnection.....	385
17.5 Python Client.....	386
17.6 Scala Client using Apache HTTP Client.....	386
17.7 Postman (Chrome Plugin) Client.....	387
Chapter 18 User Defined Functions (UDF).....	388
18.1 User Defined Functions.....	388
18.2 RuleSet Tree Recursion and Generators.....	389
18.2.1 Implementing an Extraction UDF.....	390
18.3 User Defined Variables in the UDFs.....	391
18.4 Passing input arguments in UDFs.....	391
18.5 Advanced Rule Settings in UDFs.....	391
18.6 Python code listing of Sample UDFs.....	392
18.7 Blocked Modules and Methods in UDF.....	393
18.8 UDFs in the Web UI.....	394
18.9 Supported Libraries.....	395
Chapter 19 Appendix A: Xactly.....	396
19.1 DSG with Xactly SaaS.....	396
19.2 Supported Modules.....	396
19.3 Supported Fields.....	397
19.4 Supported Profiles.....	397
19.5 Known Issues and Limitations.....	398
Chapter 20 Appendix B: Salesforce Profile.....	399
20.1 DSG with Salesforce SaaS.....	399
20.2 Supported Fields.....	399
20.3 Supported Profiles.....	400
Chapter 21 Appendix C: Standard Encoding Method List.....	401
Chapter 22 Appendix D: Known Limitations.....	406
22.1 Protegity Data Protection.....	406
22.1.1 Data Element Configuration.....	406
22.1.2 Input length.....	406
22.1.3 Null Values.....	406
22.2 Hardware Sizing.....	406
22.2.1 Memory.....	406
22.2.2 Disk space.....	407
22.3 Network.....	407
22.3.1 TLS Overhead.....	407
22.3.2 SFTP Protocol.....	407

22.3.2.1 Extended SFTP Commands.....	407
22.3.2.2 Host Key Caching.....	407
22.3.2.3 Session Negotiation.....	407
22.4 RuleSet Engine.....	407
22.4.1 XML Payload Extractor.....	408
 Chapter 23 Appendix E: Supported OpenSSL Curve Names and Options.....	409
 Chapter 24 Appendix F: Troubleshooting Data Security Gateway (DSG).....	413
24.1 Auditing and Logging.....	421
24.1.1 Forensics.....	421
24.1.2 Log Viewer.....	422
24.1.3 Audit Log Representation.....	423
24.2 Best Practices for Auditing and Logging.....	424
 Chapter 25 Appendix G: Enabling Selective Tunnel Loading on DSG Nodes.....	425
25.1 Adding a Label to a DSG Node for Selective Tunnel Loading.....	426
25.2 Removing a Label from a DSG Node for Selective Tunnel Loading.....	427
25.3 Adding a Label to a Tunnel for Selective Tunnel Loading.....	428
 Chapter 26 Appendix H: CoP Export API for deploying the CoP(Containers Only).....	430
26.1 Supported Authentication Methods for CoP API.....	430
26.2 API for Exporting the CoP Configurations.....	431
26.3 API for Exporting the CoP Configurations Using New Version.....	432
26.4 Scenarios for Exporting the CoP Configurations.....	433
 Chapter 27 Appendix I: Migrating the UDFs to Python 3.....	437

Chapter 1

Introduction to this Guide

1.1 Sections contained in this Guide

1.2 Accessing the Protegility documentation suite

The Data Security Gateway (DSG) user guide provides information about the Data Security Gateway (DSG) architecture, deployment scenarios, installing DSG on premise and cloud platforms, and working with DSG to create data protection solutions using tunnels and rules.

1.1 Sections contained in this Guide

The guide is broadly divided into the following sections:

- *Section Introduction to this Guide* defines the purpose and scope for this guide. In addition, it explains how information is organized in this guide.
- *Section Protegility Gateway Technology* introduces you to the concepts of the Protegility gateway technology and how gateway products fit into the enterprise infrastructure.
- *Section Protegility Gateway Technology Products* discusses the use cases and scenarios where the Data Security Gateway features can be leveraged.
- *Section Technical Architecture* provides the architecture overview of DSG and multiple layers that form part of the architecture. It also introduces you to key features of DSG, such as Configuration Over Programming (COP) and so on.
- *Section Deployment Scenarios* provides information about the multiple deployment scenarios for DSG.
- *Section Protegility Methodology* provides information about multiple stages of the Protegility Data Security Platform data protection process.
- *Section Planning for Gateway Installation* provides prerequisites and other configurations that must be considered before installing the DSG appliance.
- *Section Installing the DSG* provides information about installing DSG and post installation configurations. This chapter includes information about installing the DSG on-premise and on cloud platforms, namely AWS, Azure, and GCP.
- *Section Upgrading to DSG v3.1.0.5* provides information about upgrading to the latest DSG version from an older version.
- *Section DSG Web UI* introduces you to the different Web UI menus and their definitions.
- *Section Overview of Sub-Clustering* introduces you to the different Web UI menus and their definitions.
- *Section Implementation* guides you through the entire process of data protection using multiple components of DSG through a sample use case.
- *Section Transaction Metrics Logging* provides the detailed information of the operations performed by the DSG.
- *Section Error Metrics Logging* provides the detailed information about the errors that are encountered while processing a file.
- *Section Best Practices* provides information about multiple best practices that are associated with DSG components such as defining services, learn mode, and so on.
- *Section RuleSet Reference* provides information about the Ruleset elements that include the actions and payloads supported.

- [**Section DSG REST API**](#) provides an overview of DSG REST API functionality. In addition to providing a conceptual overview, the discussion steps through a use case where DSG is configured with a RuleSet object behind a REST API URL end-point.
- [**Section User Defined Functions \(UDF\)**](#) provides information about the DSG UDFs and how they can be used as API hooks that allow users to insert certain kinds of custom program logic in the DSG data processing flow.
- [**Section Appendix A: Xactly**](#) provides information about how DSG can be configured for the Xactly SaaS.
- [**Section Appendix B: Salesforce Profile**](#) provides information about how DSG can be configured for the Salesforce SaaS.
- [**Section Appendix C: Standard Encoding Method List**](#) provides a list of standard encoding methods available for as part of DSG Ruleset configuration.
- [**Section Appendix D: Known Limitations**](#) provides the known issues and limitations related to DSG.
- [**Section Appendix E: Supported OpenSSL Curve Names and Options**](#) provides information about the OpenSSL curve names and options supported by DSG.
- [**Section Appendix F: Troubleshooting Data Security Gateway \(DSG\)**](#) explains the common errors, permission restrictions, and problems you might encounter while working with DSG.
- [**Section Appendix G: Enabling Selective Tunnel Loading on DSG nodes**](#) provides information on how the Selective Tunnel Loading feature is used on the DSG.
- [**Section Appendix H: API for Exporting CoP**](#) provides information about exporting the DSG configurations through an API.
- [**Section Appendix I: Migrating the UDFs to Python 3**](#) provides information about migrating the UDFs to Python 3 version.

1.2 Accessing the Protegility documentation suite

This section describes the methods to access the *Protegility Documentation Suite* using the [*My.Protegility*](#) portal.

1.2.1 Viewing product documentation

The **Product Documentation** section under **Resources** is a repository for Protegility product documentation. The documentation for the latest product release is displayed first. The documentation is available in the HTML format and can be viewed using your browser. You can also view and download the *.pdf* files of the required product documentation.

1. Log in to the [*My.Protegility*](#) portal.
2. Click **Resources > Product Documentation**.
3. Click a product version.
The documentation appears.

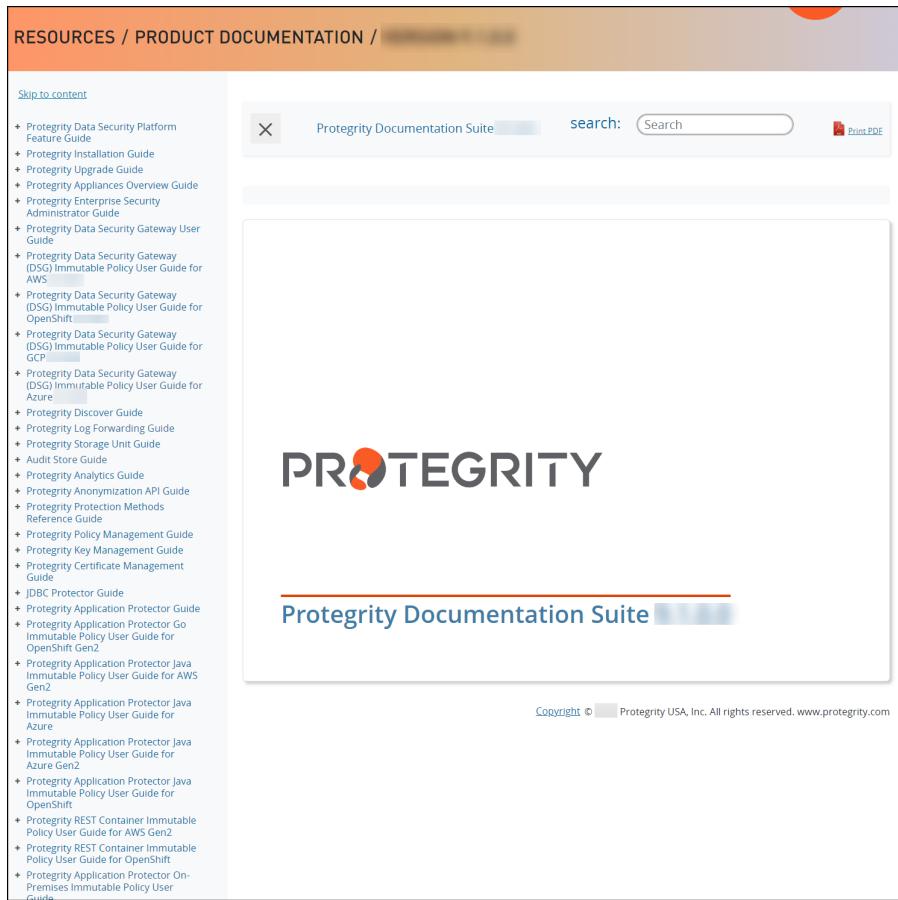


Figure 1-1: Documentation

4. Expand and click the link for the required documentation.
5. If required, then enter text in the **Search** field to search for keywords in the documentation.
The search is dynamic, and filters results while you type the text.
6. Click the **Print PDF** icon from the upper-right corner of the page.
The page with links for viewing and downloading the guides appears. You can view and print the guides that you require.

1.2.2 Downloading product documentation

This section explains the procedure to download the product documentation from the *My.Protegility* portal.

1. Click **Product Management > Explore Products**.
2. Select **Product Documentation**.
The **Explore Products** page is displayed. You can view the product documentation of various Protegility products as per their releases, containing an overview and other guidelines to use these products at ease.
3. Click **View Products** to advance to the product listing screen.
4. Click the **View** icon (ⓘ) from the **Action** column for the row marked **On-Prem** in the **Target Platform Details** column.
If you want to filter the list, then use the filters for: **OS**, **Target Platform**, and **Search** fields.
5. Click the icon for the action that you want to perform.

Chapter 2

Protegrity Gateway Technology

[2.1 Background](#)

[2.2 Audience](#)

[2.3 What is the Protegrity Gateway Technology?](#)

[2.4 How do the Protegrity Gateways Protect Data?](#)

[2.5 Why the Protegrity Gateway Technology?](#)

[2.6 Data-Centric Auditing & Protection Platform](#)

[2.7 Configuration over Programming \(CoP\) brings it all together](#)

Protegrity Gateway Technology provides an insight into the gateway technology offered that lets you protect data at rest as well as on the fly.

2.1 Background

The most important asset for organizations today is data. Data is being collected at an unprecedented rate. Data analysts and data mining scientists develop analytical processes to gain transformative insights from the collected data to gain corporate advantages, growth, and innovation.

This rich pool of data is commonly tied to individuals, such as employees, customers, patients, and the like, making it a target for identity theft. The ever-increasing cases of data breaches is a proof that the business of stealing data is a large and lucrative business for hackers. In effort to stop data thefts, organizations are constantly looking for innovative solutions for protecting sensitive data without affecting the use and analysis of this data.

2.2 Audience

Multiple stakeholders collaborate to deliver enterprise level data security solutions. Some are responsible for setting corporate business requirements while others own the responsibility of designing and implementing data security solutions.

The audience for this document is the following stakeholders who play a role in the data security ecosystem:

- **Business Owners:** Focused on maximizing the value and growth delivered by their business system. Data security concerns and security solutions may prevent business owners from executing their plans. These stakeholders are the advocate for the data and its untapped potential.
- **Security Professionals (CISO, Security Officers, Governance, Risk, etc.):** Responsible for keeping business systems secure. They must understand the goals of the business owners and design and deliver data security solutions that offer a balance between protecting the data and enabling business usage. These security professionals:
 - Set the security risk tolerance for the organization
 - Identify the data that is deemed sensitive in an organization
 - Design and implement the data security solution that meets business requirements
 - Establish the ongoing monitoring and alerting of sensitive data

- **IT (DBA's, Developers, etc.):** Responsible for implementing and deploying business and data security solution. Some organizations have a specialized IT team that is part of the security organization. In this document, this team is identified as the team that implements and deploys the data security solution, irrespective of their location in the organization chart.
- **System Architects:** Equipped with deep knowledge of their own business infrastructure and of the corporate data security requirements makes them the center authority responsible for the technical architecture of the data security solution.

These stakeholders are involved from the initial stages of vetting data security vendors to the eventual design of the data security architecture implemented by the IT stakeholders.

2.3 What is the Protegility Gateway Technology?

Protegility Gateway Technology is an umbrella term for the new and innovative push to deliver data security solutions from Protegility that is highly transparent to corporate infrastructures.

When adopting data security solutions, companies expect minimal impact on existing business systems and processes. In the past, data security solutions have been integrated into business applications and databases.

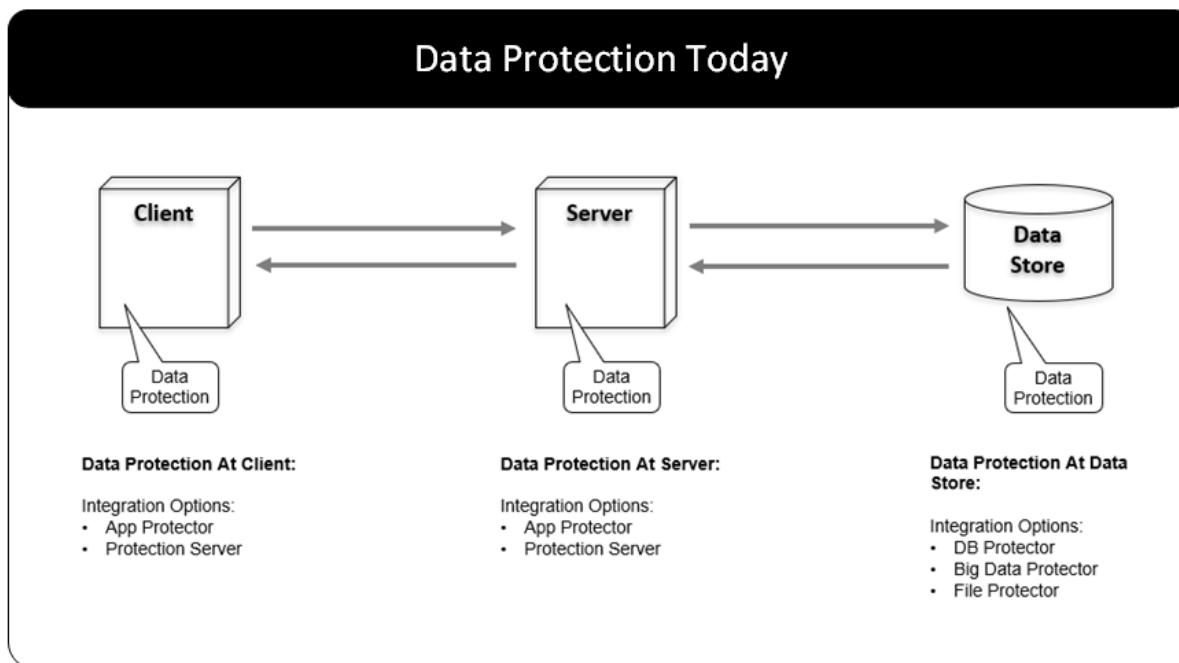


Figure 2-1: Current Data Protection Scenario

These approaches require changes to these systems.

The gateway is a network intermediary between systems that communicate with each other through the network. By delivering data security solutions on the network, changes to the existing systems are avoided or minimized.

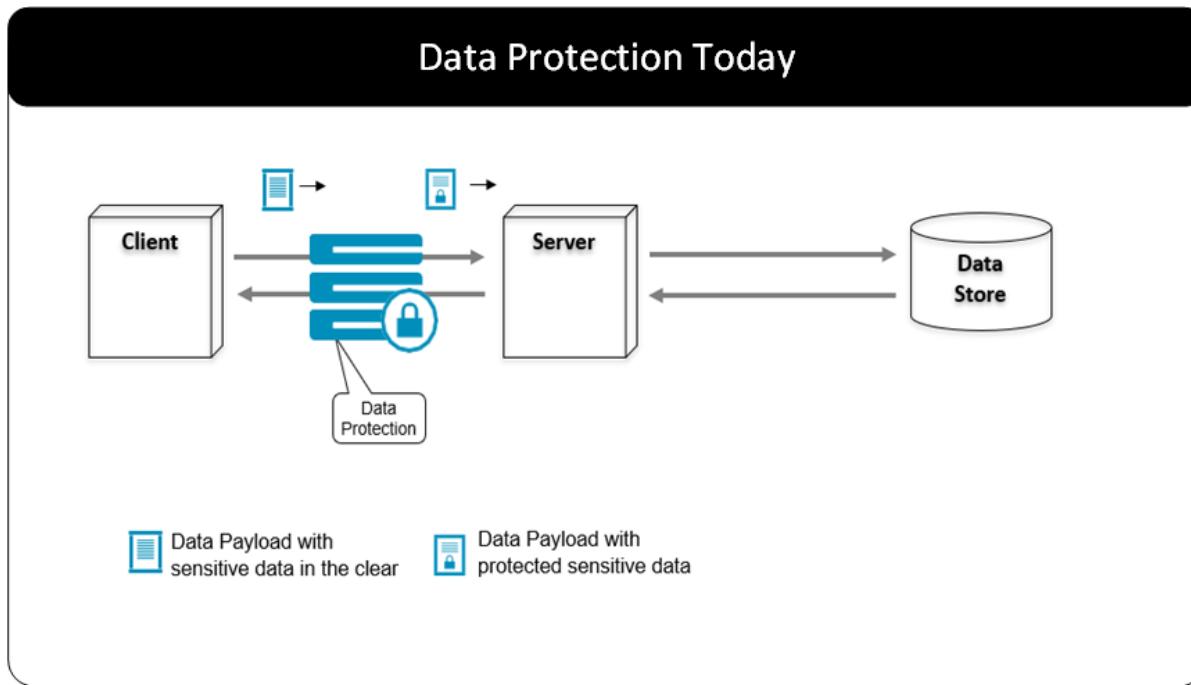


Figure 2-2: Data Protection Scenario with Intermediary

The Protegility Gateway Technology represents a set of products that deliver data protection on the network. These products include:

- Protegility Cloud Security Gateway
- Protegility Data Security Gateway

This document focuses on the Cloud Security Gateway and the Data Security Gateway.

2.4 How do the Protegility Gateways Protect Data?

Protegility gateways deliver security operations on sensitive data by peering into the payloads that are being transmitted through the network.

The gateway intercepts standard protocols such as TCP/IP, scans the payloads carried on the backs of these protocols for sensitive data, and applies security operations (protection or un-protection) of the sensitive data as it passes through the gateway.

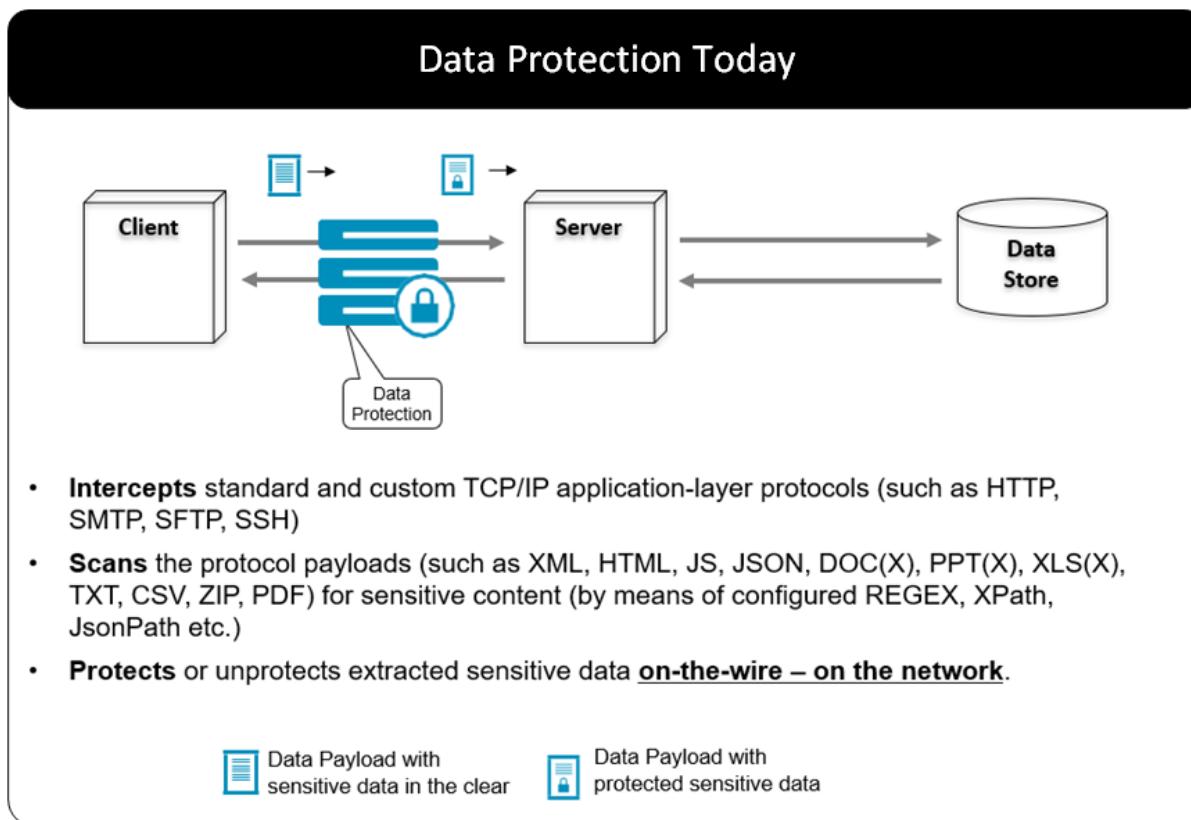


Figure 2-3: Data Protection Scenario with Intermediary process

With the gateway approach to delivering security operations, impact to existing systems is eliminated or minimized.

2.5 Why the Protegility Gateway Technology?

The Protegility Gateway Technology represents an extension to the Protegility Data-Centric Audit and Protection (DCAP) platform and Protegility Vaultless Tokenization that are being used today to protect sensitive data across the largest enterprises throughout the world.

The combination of the Protegility DCAP platform, Protegility Vaultless Tokenization, and the Protegility Gateway Technology delivers many benefits:

- **Enterprise:** As yet another protector in the Protegility Data Centric Audit and Protection platform family, the Protegility Gateways can receive and use policies from the Enterprise Security Administrator (ESA) that define rules for how you want to protect sensitive data throughout your enterprise. The protected data is interoperable with other protectors in the DCAP family.
- **Transparent:** Delivering data protection on the network eliminates the need to modify the source or destination systems. This makes the implementation of a security solution easier than if you had to modify application code or database schema.
- **Fast:** The gateway provides the fastest mechanism to protect and unprotect data at line speed. The granularity of the security operations is very high since the operations are applied very close to the data with no latency.
- **Scalability:** The gateways can scale vertically as well as horizontally. The vertical scaling is enabled through the addition of CPU and RAM while horizontal scaling is enabled by adding more nodes to a gateway cluster.
- **Configuration over Programming (CoP):** The implementation of data security with the Protegility Gateways does not require programming. Implementation is configured through an easy to use web interface. This practice is called Configuration over Programming (CoP).
- **Deployment Flexibility:** The Protegility Gateways as well as the Protegility DCAP platform can be deployed on-premise, in the cloud deployment, or in a hybrid deployment architecture.

- Use Cases:** The Protegility Gateway Technology is a kind of “Swiss Army Knife” for applying data security across many use cases that are described in detailed in this document.
- Extensibility:** While CoP delivers virtually all you need to implement data security solutions using the Protegility Gateways, you may extend the functionality using Python programming language through User Defined Functions.
- SaaS Protection Agility:** SaaS applications (Salesforce, Workday, Box, etc.) have gained popularity due to the ease with which you can add business functionality to your organization. Because of their cloud-based deployment model, SaaS applications can change quickly. The approach of implementing data security solutions with Configuration over Programming (CoP) makes it easy to keep up with these changes and avoid outages.

2.6 Data-Centric Auditing & Protection Platform

The Protegility Gateway Technology is part of the Protegility Data-Centric Audit and Protection family of protector, which together with the Enterprise Security Administrator (ESA) makes up the Protegility Data-Centric Audit and Protection Platform.

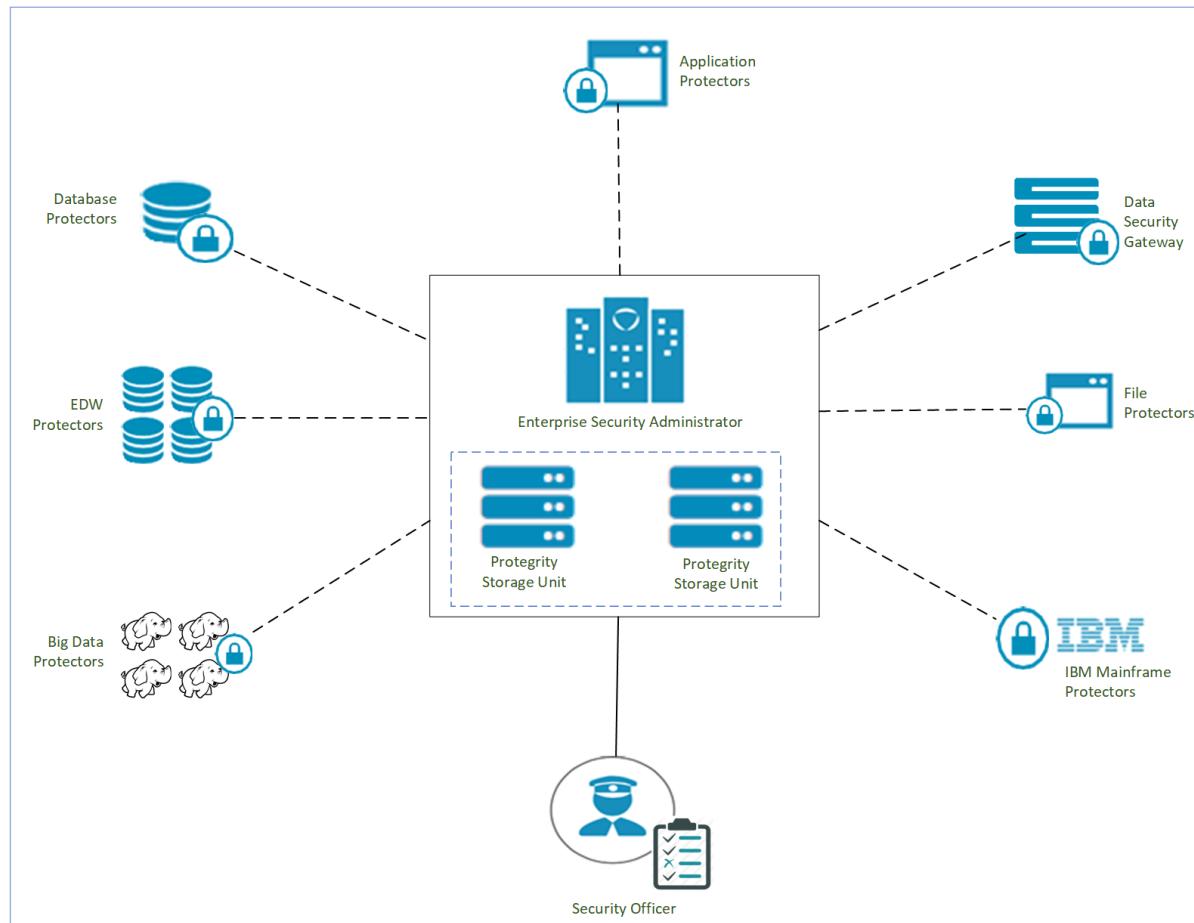


Figure 2-4: Protegility Data-Centric Audit and Protection Platform

The Enterprise Security Administrator (ESA) is a central point of management of data security policies enforced by various protectors. Each protector is designed to accept an ESA policy that provide the rules for protecting and unprotecting sensitive data.

Security operations on sensitive data performed by any of the protectors can be audited. Audit logs based on invoked security operations are sent back to ESA for reporting and alerting purposes.

You can protect sensitive data in any business system that is secured with Protegility, let the protected data travel in a protected state between different business systems, and then unprotect this data in a target system for authorized users.

Fine Grained and Coarse Grained Data Security:

Data Security can span many different types of security that are differentiated by based on the point where security policies are applied. Security can be applied to the perimeter by only letting some users in.

Security applied to data can be delivered in the following forms:

- It can be applied to the data as an access control layer by hiding data from un-authorized users.
- It can be applied to data by encrypting the raw storage associated with data stores. This is sometimes called coarse grained data protection or tablespace protection or Transparent Database Encryption.
- It can also be applied to the specific data itself with encryption or tokenization such as the Social Security Number, the E-mail address, the Name and so on.

Protegility Vaultless Tokenization enables you to reduce the scope of systems where sensitive data exists in the clear, with minimum to no impact of its business usage.

Together with other protectors contained in the Protegility Data-Centric Audit and Protection platform, the Protegility Gateway Technology products deliver on these approaches with a single product.

2.7 Configuration over Programming (CoP) brings it all together

The Protegility Gateway Technology brings together the ability to peer into networks payloads and the security policy-based rules for protecting sensitive data with a concept called Configuration over Programming (CoP).

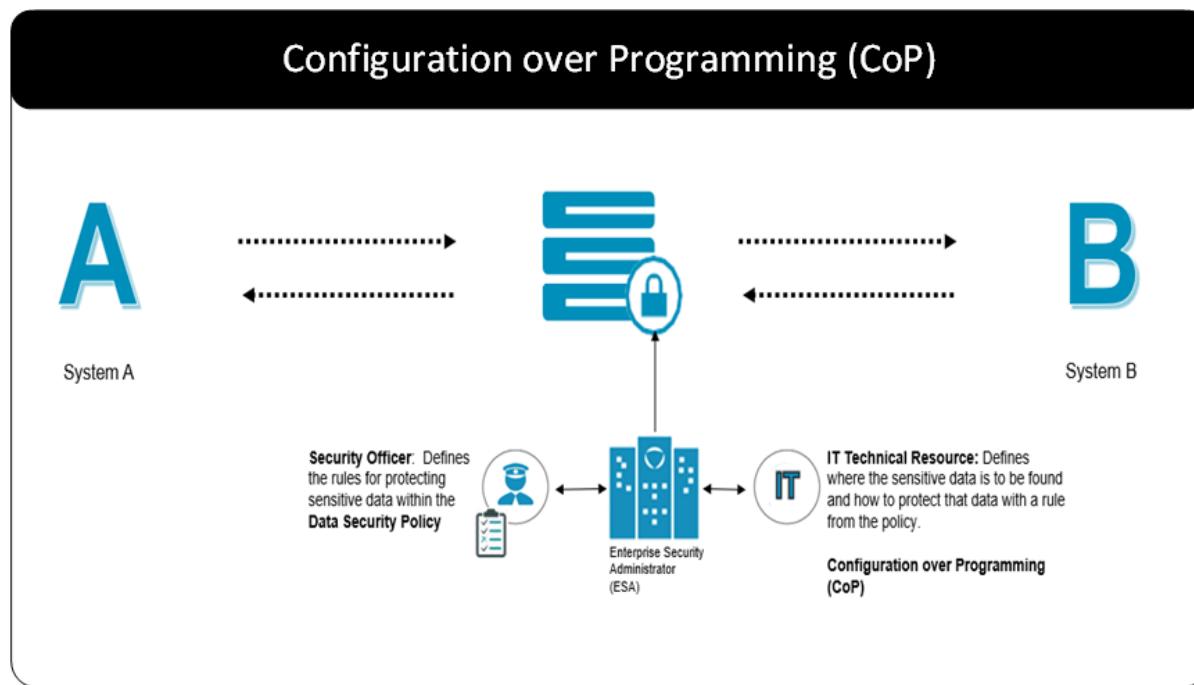


Figure 2-5: CoP Concept

The diagram above depicts the gateway along with the components that together constitute CoP. Data is transmitted bidirectional between System A and System B. The Gateway acts as a network intermediary through which the transmissions pass. A set of rules called CoP Profiles define the transformations performed on that data.

Security Officers set rules that define how the security team would like corporate security solutions to treat sensitive data. Having the security team define these rules across the corporate data asset delivers consistency in the security treatment. This helps both security and the usability of the data. Having the security team responsible for this task also delivers separation of duties that

reduce or eliminate a conflict of interest in who sets the rules for protecting sensitive data and who can see the sensitive data in the clear.

IT Technical Resources bring it all together through CoP. CoP enables a technical resource, a CoP Administrator, to create a set of CoP Profiles that blend the different aspects of delivering security or other transformations on data. A CoP Profile include:

- **Data Collection:** The data collection profile rules define the network protocols that are being inspected. For example, you can instruct the gateway that it will inspect HTTP or SFTP. These are standard protocols on top of which the transmission of data across networks are built.
- **Extend Gateway:** If you have a custom protocol, the gateway can be extended and configured to accept that as well.
- **Data Extraction:** Protocols carry many kinds of payloads that are commonly used. They carry HTML payloads that define web pages, PDFs that contain document content, JSON documents that are used to transfer business system data from one application to another, and many others. CoP Profiles are configured to identify specific data within these commonly used payloads.
- **Extend Codecs:** A set of extraction codecs are included with the gateway but as with the protocols, the data extraction codecs can also be extended to include new standard or custom codecs.
- **Actions on Data:** Once you have defined the protocol and identified specific data within a payload, you can then apply an action or a transformation on the data. In the context to data security, this action is a security operation (protect or unprotect) where the rules for performing that security action come from the policy rules identified by the security team.

However, other actions can also be defined with User Defined Functions that enable the CoP Administrator to define whatever transformation is needed for a scenario.

The CoP Profile is defined with an easy to use interface that does not require programming. However, you can extend the collection, extraction, and actions beyond what is available out of the box. This powerful product and the CoP form of configuring its behavior to meet your business needs can go well beyond security in its application.

Chapter 3

Protegrity Gateway Technology Products

3.1 Cloud Security Gateway

3.2 Data Security Gateway

The Protegrity Gateway Technology consists of several gateway products including the Cloud Security Gateway and the Data Security Gateway.

These products are based on applying security operations on the network. This document focuses on the Cloud Security Gateway and the Data Security Gateway.

Data Security Gateway (DSG) offers flexibility and can run in the following environments:

- **On-premise:** The appliance is installed and runs on dedicated hardware.
- **Virtualized:** The appliance is installed and runs on a virtual machine.
- **Cloud:** The appliance is installed and runs on or as part of a Cloud-based service.

3.1 Cloud Security Gateway

SaaS based business applications are being adopted at a rapid pace. Organizations use SaaS-based application to easily fulfill different business needs. For example, SaaS based CRM applications can be used to manage their customer relationship. Company's purchasing SaaS applications are outsourcing the burden of development, maintenance, and infrastructure to the SaaS vendors. All you need is a subscription contract and a browser.

These SaaS applications store corporate data in the cloud and some of this data may be sensitive. The Cloud Security Gateway can be used to protect sensitive data as it moves from a corporate user on a browser to the SaaS application storage on the cloud. When the data is returned to the user, the sensitive protected data is unprotected and delivered to the intended user in a usable form.

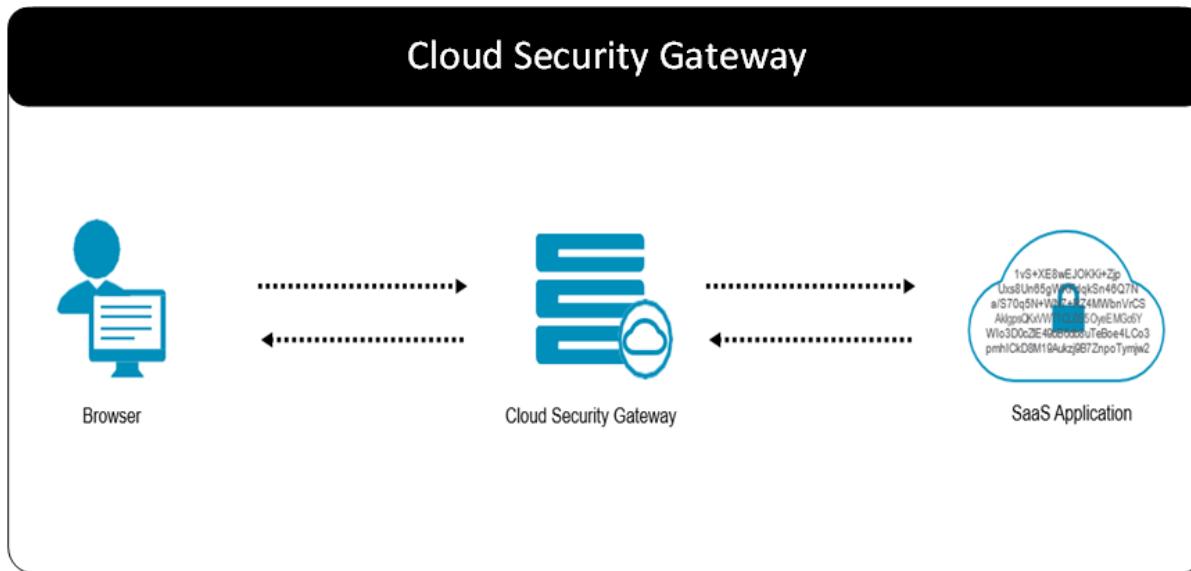


Figure 3-1: Cloud Security Gateway with SaaS Application

CoP Profiles are available for some SaaS applications or they can be created to configure the Cloud Security Gateway to protect and unprotect sensitive data for a specific SaaS application. The Cloud Security Gateway delivers differentiation from other vendors described in the following points:

- **Security:** When the Cloud Security Gateway protects sensitive data that is stored in the SaaS cloud storage, this data is it's rendered in an unusable form to system administrators who maintain the SaaS application. You are also protected from the bad guys if they breach the security imposed by the SaaS application.
- The cryptographic keys that are used to protect and unprotect your sensitive data are stored on your premise, giving you complete control over your sensitive data.
- The Protegility Cloud Security Platform can be used for Data Residency Applications that have been recently highlighted by the changes in the European safe harbor requirements.
- **Enterprise Data Security:** CoP Profiles use the data security policy rules created by security professionals to control the how sensitive data is protected throughout an organization. These rules are created in the centrally administered Enterprise Security Administrator (ESA). Any data that is protected in the SaaS and that moves to or from other business systems will realize the benefit of a consistent enterprise data security model. Secured data is interoperable between business systems. Data is then secured at rest and in transit.
- **Agility:** SaaS applications change since the SaaS model on top of which they are built enables SaaS companies to add functionality very easily with no burden of installing new software or other complexities. They simply develop a new feature and that feature is available the next time a user brings up the SaaS application on their browser.
- Data security solutions that protect SaaS applications must be agile in reacting to these changes. Some SaaS Data Security vendors require building new products to accommodate these changes in their security solution. The Protegility Cloud Security Gateway uses the Configuration over Programming (CoP) model to make changes rapidly through CoP Profiles that require no programming, just configuration.
- **SaaS Change Monitoring:** Companies that buy subscriptions to SaaS applications do so because it lets them add business functionality to their business easier and faster than ever before. So, when these SaaS applications change, they expect that these changes will not break the security solution. Protegility will maintain systems that track these SaaS changes and test against the CoP Profiles that have been created for a specific SaaS. These changes will be identified and CoP Profile configuration modifications will be provided to keep your business systems secure and running.
- **Pricing Model:** Protegility delivers the Cloud Security Gateway can be purchased with the same model that is used to purchase the SaaS applications themselves to make the process of adding security to your SaaS applications as seamless as possible.
- **3rd Party Integration:** SaaS applications will communicate with other SaaS applications and business processes through APIs often in the form of RESTful APIs. The Cloud Security Gateway can be used to perform security operations between these systems. For example, you may protect your data in a SaaS application like Salesforce. When data is pulled from

Salesforce into a 3rd party application or into other business systems, the data may need to be in the clear (unprotected). The Cloud Security Gateway can intercept these API's and unprotect data that needs to be used in the 3rd party business process.

3.2 Data Security Gateway

The Data Security Gateway applies data security operations on the network and uses the CoP Profiles to configure a data security solution.

When building modern business applications, companies use browsers as the primary means of user interaction between users and the business system. Network based protocols such as web services or RESTful APIs are also the standard way of calling out to execute a business function or as the way to communicate between systems.

Flexibility: The Protegility Data Security Gateway can be applied to several use cases that are outside the realm of SaaS applications. This “Swiss Army Knife” type of product can single handedly account for different data security scenarios that together will constitute a holistic and complete solution. This is a powerful addition to your development set of tools.

Transparency: Most data security solutions today require integration with your host business system components such as applications and databases. By applying security operations on the network, the Data Security Gateway reduces the amount of work that you must do to protect your organization.

Enterprise: Data protection with the Data Security Gateway can be combined with any other protector in the Protegility Data-Centric Audit and Protection platform. If the gateway does not meet a specific requirement, you will be able to fulfill your data security solution with any one of the other protectors in the platform.

The set of security scenarios described in the following section will give you a glimpse of the flexibility of this product.

3.2.1 Data Security Gateway Scenarios

The Data Security Gateway can be implemented for multiple scenarios, which are listed in this section.

3.2.1.1 Data Security Gateway used to protect Web Applications

Most applications today are based on a web interface. Web interfaces have an architecture that protect data from the browser to the Web Server with HTTPS. The Web Server terminates HTTPS and the traffic flows through Application Servers and ultimately into a data store. Sometimes there are business systems that process data from the databases.

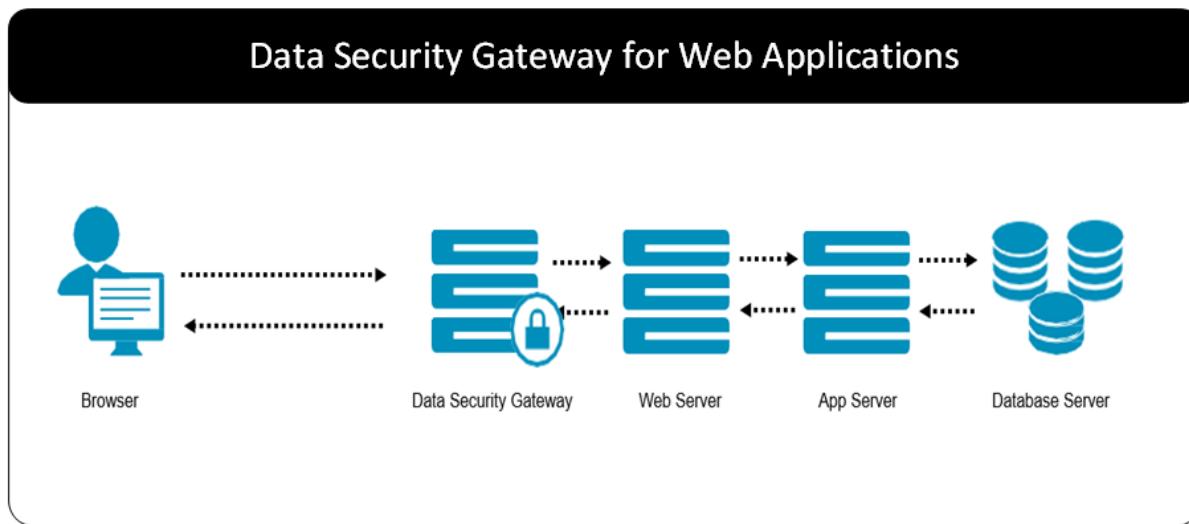


Figure 3-2: Data Security Gateway for Web Applications

From a security point of view, the data that is flowing after HTTPS is terminated will be in the clear. By adding the Data Security Gateway before the Web Server, the Data Security Gateway can terminate HTTPS and execute the CoP Profile to extract and

protect any sensitive data that may be in the payload. For example, if SSN is entered in a web form, a CoP Profile rule can be created that picks out the HTML label associated with the value entered and protect the SSN.

Like Web Servers and App Servers, the Data Security Gateway are stateless and can be delivered behind a load balancer. Vertical and Horizontal scaling can handle whatever throughput is required to maintain business performance.

If business functions are performed in the App Server on sensitive data or in the database itself within stored procedures, you can call out to the RESTful API Server that can be also provided by the Data Security gateway or by using database UDFs from the Protegility Database Protector. You have many options.

3.2.1.2 Data Security Gateway as an API Security Gateway

API's are the interoperability language of modern architectures. In addition to the classic use of APIs for interfaces internal to applications, APIs are now moving data and executing processes between disparity applications.

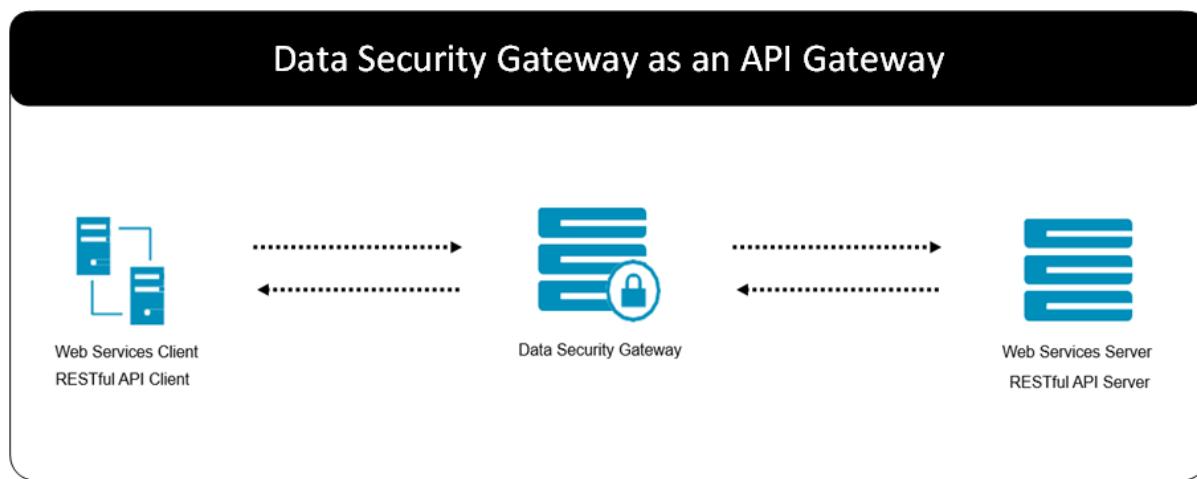


Figure 3-3: Data Security Gateway as an API Gateway

These APIs are implemented in the form of SOAP based Web Services or RESTful APIs. In these scenarios you may have used either of these techniques to implement both the client and the server component. The server component may also come from different vendors.

Irrespective of how you implemented this architecture, the Data Security Gateway can be placed between the client and the server where a CoP Profile can be used to specify what and how sensitive data passing in either direction will be protected or un-protected.

3.2.1.3 Data Security Gateway for Files

A common scenario found in enterprises today is the delivery of files into an organization from business partners. Oftentimes, the security approach to protecting data in transit from the source to the destination is SFTP.

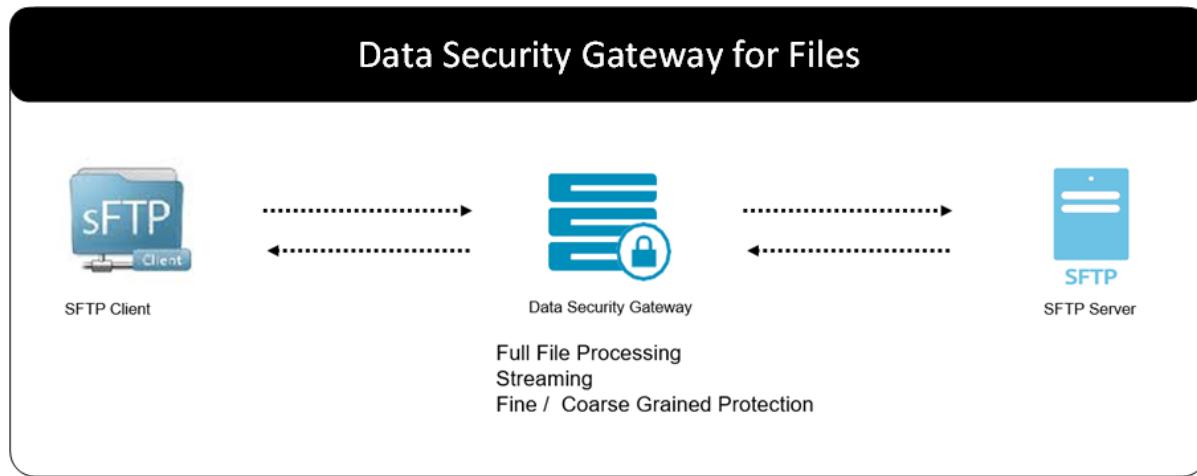


Figure 3-4: Data Security Gateway for Files

The problem is that when the file lands in the SFTP Server, the encryption is terminated and the sensitive fields are now exposed in an unprotected form. This is like the Web Application scenario where HTTPS is terminated by a Web Server and all sensitive data downstream is exposed in the clear.

When the Data Security Gateway is placed before the file lands on the SFTP Server the gateway can terminate security making the payload visible to the gateway. A CoP Profile can be created to perform different types of security operations to the file.

Coarse Grained Protection: CoP Profile rules can be created to encrypt the entire file before it lands on the SFTP Server.

Fine Grained Protection: CoP Profile rules can be created to encrypt or tokenize specific data elements contained within the file.

3.2.1.4 Data Security Gateway as an On Demand RESTful API Server

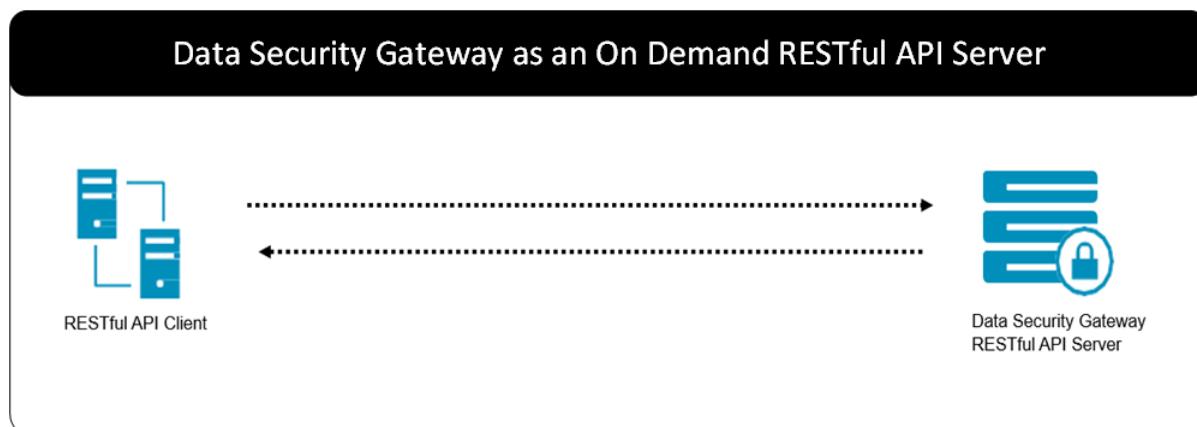


Figure 3-5: Data Security Gateway as an On Demand Restful API Server

Protecting or unprotecting data is easy. You simply send your XML or JSON document that contains the data to be acted on and the CoP Profile rules take care of the rest. The rule will identify the exact part of the document and act on that data. There is no need to parse that data out and then reconstruct the document after the security operation is performed.

Chapter 4

Technical Architecture

[4.1 System Architecture](#)

[4.2 Configuration over Programming \(CoP\) Architecture](#)

[4.3 Codebook Re-shuffling](#)

This section provides an overview of the Data Security Gateway (DSG) technical architecture and key concepts related to it.

4.1 System Architecture

Protegility Gateway Technology products are assembled on a layered architecture.

The lower layers provide the foundational aspects of the system such as clustering and protocol stacks.

The higher layers are specialized and provide various business functions or building blocks that are used to create Rulesets or instruction on how the gateway should act on data. Some of these building blocks include functions such as decoders for various data formats as well as data transformation for cryptography.

The gateway architecture is unique in a way that while it provides standard out-of-the-box building blocks, these building blocks are extensible at each layer by the customer with building blocks that are relevant to the customer requirements. These requirements don't have to be security requirements, they can be any requirements that will aid the customer in processing their data.

The following figure shows a view of the gateway system architecture.

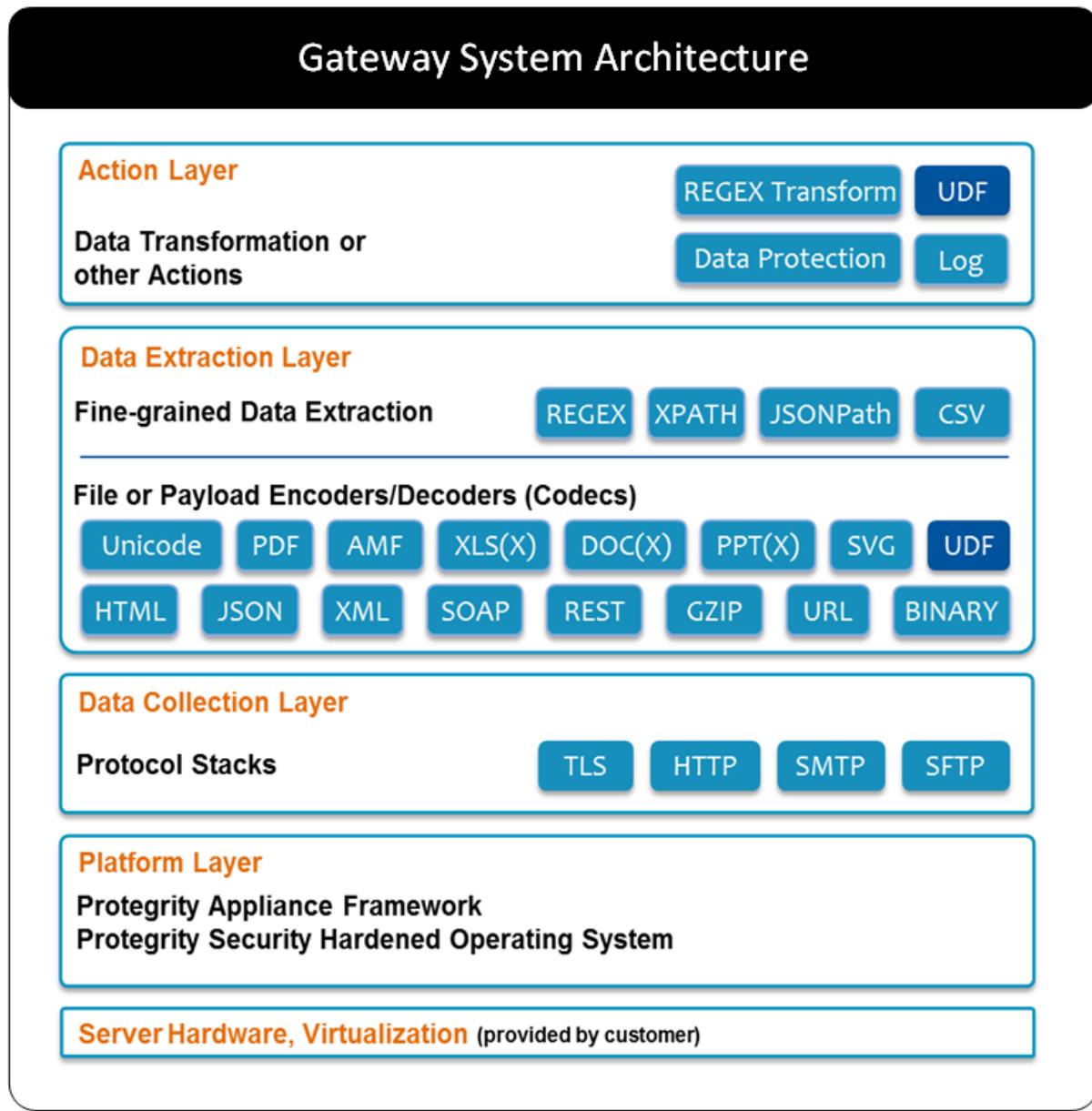


Figure 4-1: Gateway System Architecture

4.1.1 Platform Layer

The Platform Layer runs on top of customer-provided hardware or virtualization resources. It includes an operating system that has been security-hardened by Protegility and an infrastructural layer running above it called the Protegility Appliance Framework. The Protegility Appliance Framework is responsible for common services, such as inter-node communications mechanisms and clustering.

Data communicated through the platform layer is passed onto the Data Collection Layer for further processing.

4.1.2 Data Collection Layer

The Data Collection Layer is the glue between the higher layers of the gateway and the external world. It is responsible for ingesting data into the gateway and passing it on higher layers for further processing. Likewise, it is responsible for receiving data from the higher layers and outputting it to the external world.

In terms of the layered TCP/IP model of networking, this is the transport/application protocol layer of the gateway architecture.

Since the primary method through which gateway interfaces with external world is over networking, data is typically delivered to and from the gateway by means of application-layer protocols such HTTP, SFTP and SMTP. The gateway terminates these protocol stacks.

These protocols can be extended to any protocol that a company has created for their own requirements by creating a custom protocol using the gateways' User Defined Function (UDF).

Data delivered through these protocols are passed to the Data Extraction Layer for further processing.

4.1.3 Data Extraction Layer

The Data Extraction Layer is at the heart of fine-grained data inspection capabilities of the gateway.

The Data Extraction layer is split into two logical functions:

- **Codecs:** These are the parsers or the data encoders and decoders targeted at individual native formats such as XML, JSON, PDF, ZIP, and Open-Office file formats such as DOCX, PPTX, and XLSX.
- **Extractors:** These are responsible for fine-grained extraction of select data from within the larger bodies of data outputted by the codec components. These include mechanisms such as Regular Expressions, XPath, and JSONPath.

The subsets of data extracted by the Data Extraction Layer are passed up to the Action Layer, where they may be transformed for data security or acted upon for some other business logic. Transformed data subsets received from the Action Layer are substituted in their original place in the original payload. The modified payload is encoded and delivered down to the Data Collection layer for outputting to the external world.

The building blocks in this layer can be extended to include your custom requirements through User Defined Functions (UDFs). UDFs enables customers to build and extend the gateway with their own data decoding and extraction logic using Python programming language.

Data extracted from payloads is passed to the Action Layer for further processing.

4.1.4 Action Layer

The Action Layer is responsible for operating upon the data passed on to it by the Data Extraction Layer. The data extracted is acted upon by actions in the Action Layer.

Operating on this data may include transforming the data for security purposes. This includes all of the data security capabilities offered by the core Protegility data security platform, such as protection by means of encryption or tokenization, un-protection, re-protection, hashing, and masking.

This layer also includes a UDF component, which enables customers to extend the system with their own action transformation logic using Python programming language.

4.2 Configuration over Programming (CoP) Architecture

This section provides detailed information about Configuration over Programming (CoP) concepts and how creating rules is made an easy process due to this paradigm shift.

4.2.1 CoP Overview

Configuration over Programming (CoP) is a key paradigm used in the Protegility Gateway Technology. The CoP technology enables a CoP Administrator to create a set of rules that will instruct the gateway on how to process data that traverses it.

The CoP technology is also a key technology from a user experience point of view. The structure of the rules is just as important as the rules themselves. The set of rules, their structure, and an easy to use interface results in a powerful concept called Configuration over Programming (CoP).

The Data Security Gateway (DSG) is fundamentally architected on the CoP principle. CoP suggests that configuration should be the preferred way of extending or customizing a system as opposed to programming. Users configure Rules in DSG WebUI to define step-by-step processing of incoming messages. This allows DSG users to handle any kind of input message (e.g. CSV, XML, JSON, fixed-width, free-form text) so long as they have corresponding rules configured in the DSG. The rules are generally categorized as – extraction (e.g. message parsing) and transformation (e.g. data protection).

The DSG product evolution started with Static CoP wherein the request processing rules are configured ahead of time. However, DSG now also offers Dynamic CoP that allows rule definitions (JSON structured) to be dynamically injected in the request messages (e.g. as an HTTP header field) and executed on the fly.

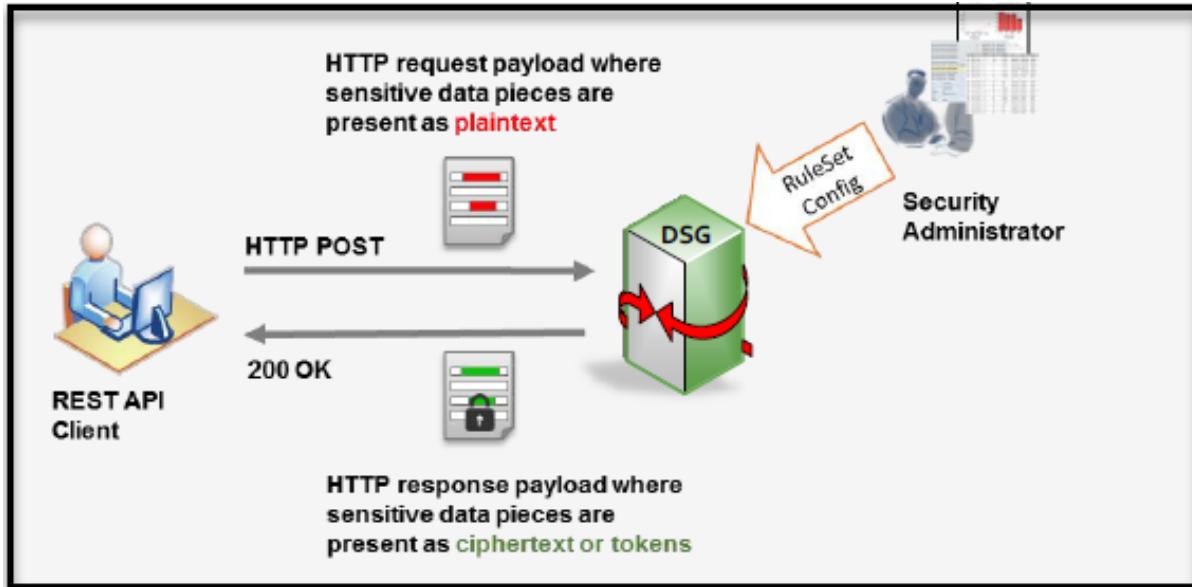


Figure 4-2: Static CoP

DSG users configure the CoP RuleSets to construct a REST API that is suitable to their environment. DSG's RESTful Interface is fairly high-level such that its API users are not exposed to low-level underlying crypto API message sequences such as open and close session. Further, low-level parameters such as data element name, session handle etc. are not exposed either. User identity can either be pre-configured in DSG, derived as a result of HTTP basic authentication user or dynamically provided through the API as an HTTP header (the name of the header is user configurable) or some part of the HTTP message body.

The following figure shows high-level functionality of the DSG RESTful interface.

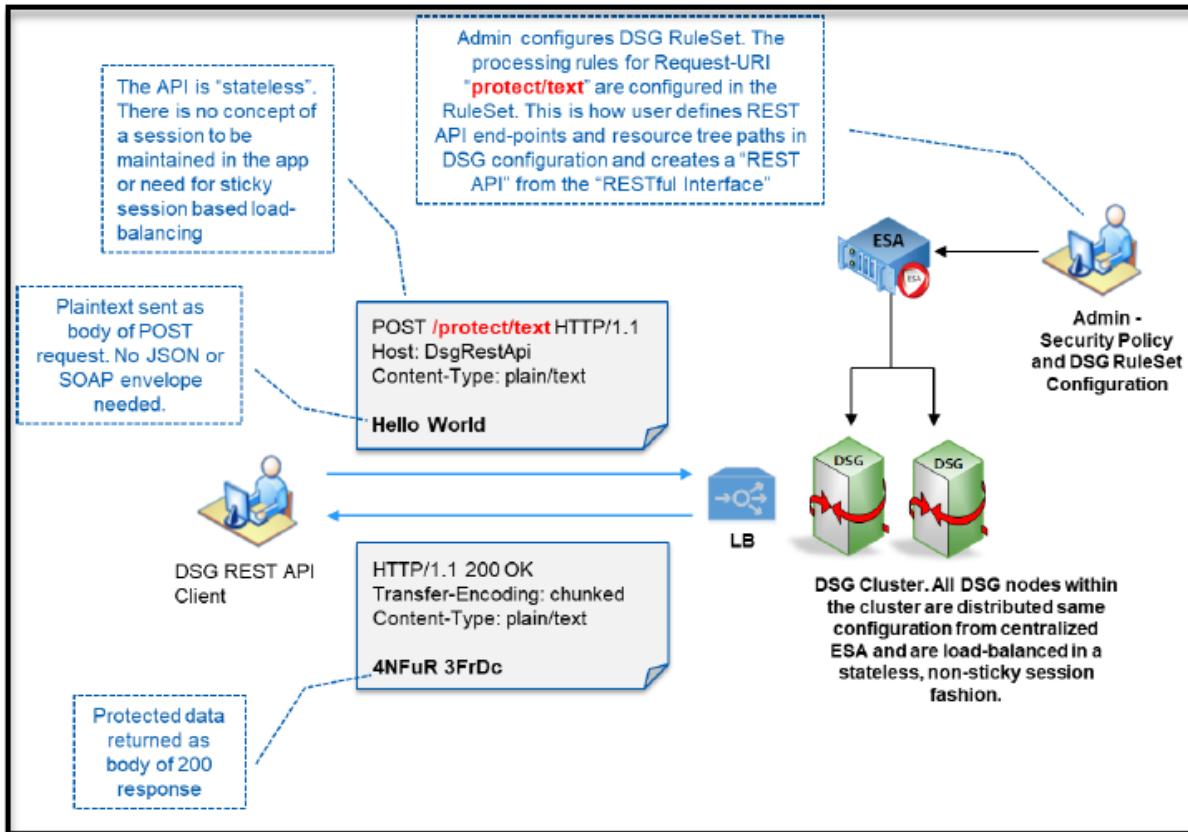


Figure 4-3: DSG REST API framework with Static CoP

Note that for simplicity the DSG example above shows a simple text string that is being tokenized word-by-word with the tokens being returned in the 200 OK response. In reality, DSG comes with a whole battery of “codecs”. Codecs are message parsers that allow DSG to parse and process complex payload bodies. DSG’s codecs include – XML, JSON, Text, Binary, CSV, Fixed-Width, MS-Office, PDF, Google Protocol Buffers, HPE ArcSight CEF, Date-Time, and PGP. Further, DSG allows custom extraction and transformation rules to be written in Python and plugged-in within DSG CoP RuleSets.

The following sections describe Ruleset, the Ruleset Structure and the Ruleset engine followed by an example.

4.2.2 CoP Ruleset

As described in the System Architecture section, the Protegility Gateway Technology contains built-in standard protocol codecs which allow configuration-driven payload parsing and processing for most data security use cases seen in typical networking protocols.

The Ruleset describes a set of instructions that the gateway uses to transform data as it traverses the gateway in any direction.

The different kinds of Rule objects (instructions) currently available in the gateway are shown and described in the following figure.

Types of Rule Objects in a Ruleset

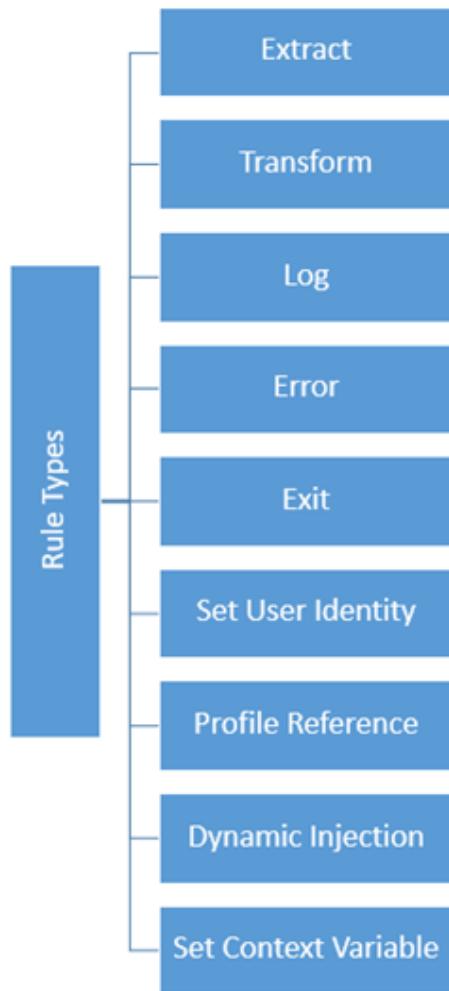


Figure 4-4: Rule Objects in a Ruleset

A typical Ruleset is constructed from both Extract and Transform rules.

The core rules available today are:

- **Extract:** As the name implies, Extraction rules are responsible for extracting smaller pieces of data from larger bodies of data. By way of engaging existing codecs, they are also capable of interpreting data per predefined encoding schemes. While the Extraction rules act as data filters, they do not actually manipulate data. Therefore, they are branch nodes in Ruleset tree and have children rules below them.
- **Transform:** Transformation rules are responsible for manipulating data passed into them in some way. Typical data security use cases will employ Transformation rules for performing data protection, un-protection, re-protection, masking or hashing which are pre-packaged in the gateway.

When customers require transformations that may not warrant out-of-the-box security actions, they can build their own actions with Transformation User Defined Functions (UDFs). Customers can extend the out-of-the-box transformations with UDFs.

- **Log:** The Log rule object lets you add log entries to the DSG log. You can define the level of logging that needs to be reflected in the log, such as warning, error, etc. The decision of where the log needs to be saved, as internal log that can be accessed through Log Viewer or Forensics, can also be made in this rule.
- **Exit:** The Exit option acts as a terminating action and the rules are not processed further.
- **Set User identity:** The Set User Identity rule object comes in effect if user name details are part of the payload. The Protegility Data Protection transformation leverages the value set in this rule such that the subsequent transformation actions calls are performed by this set user.
- **Profile Reference:** You can refer to an external profile using the Profile Reference action. This rule transfers the control to a separate batch of rules grouped in a profile.
- **Error:** You can use this action to add custom response message for any invalid content.
- **Dynamic Injection:** You can use Dynamic CoP to send rules for extraction and transformation as part of a request header along with the data for protection in request message body.
- **Set Context Variable:** You can use this action type when you want to pass any value as an input to the rule. The value set due to this rule will be maintained throughout the rule lifecycle.

4.2.3 Ruleset Structure

Rulesets are organized in a hierarchical structure where Extract rules are branch nodes and other rules such as Transform rules are leaf nodes. In other words, you want to Extract some data from the payload and then perform a Transform action on the data extracted.

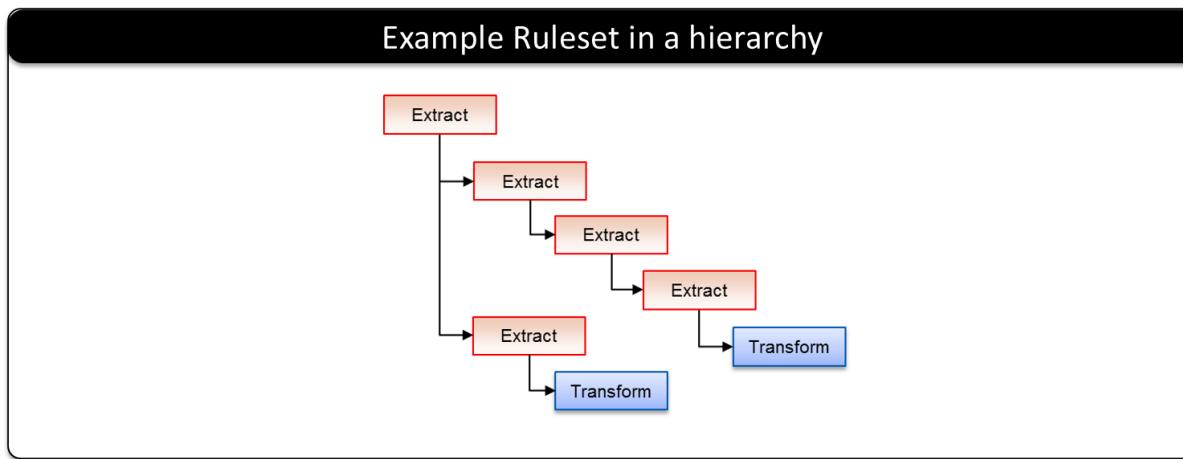


Figure 4-5: Example Ruleset Hierarchy

Rules are compartmentalized into Profile containers. Profile containers can be enabled or disabled and they can also be referenced by a Profile Reference rule.

4.2.4 Ruleset Tree of Trees (ToT)

A typical Ruleset are recursed and processed in a sequence. With this mechanism, sibling rules that belong a given parent and all the child rules that belong to a sibling rule are recursed and executed sequentially from top to bottom with no provision for conditional branching.

A fundamental drawback of this mechanism is that it disallows decision-based, mutually-exclusive execution of individual child rules on different parts of extracted data within the same extraction context, such as a row within a .CSV file, groups within a regular expression, or multiple XPaths within an XML document. This leads to extraction or parsing of the same data multiple times because different parts of extracted data within the same extraction context needs to be processed differently, such as protection using different data elements, and so on.

The RuleSet Tree of Trees (ToT) feature is an enhancement to the RuleSet algorithm that addresses this drawback. With the RuleSet ToT feature, an extraction parent rule can have multiple child rules that can be executed mutually-exclusive to each other based on some condition applied in the parent rule. The feature allows different parts of extracted data to be processed downstream using different profile references. Since the profile references are sub-trees in and of themselves, this feature adds a Tree-of-Trees structural notation to the CoP RuleSets.

The following compares the layout and execution paths of traditional RuleSets with the ToT RuleSets:

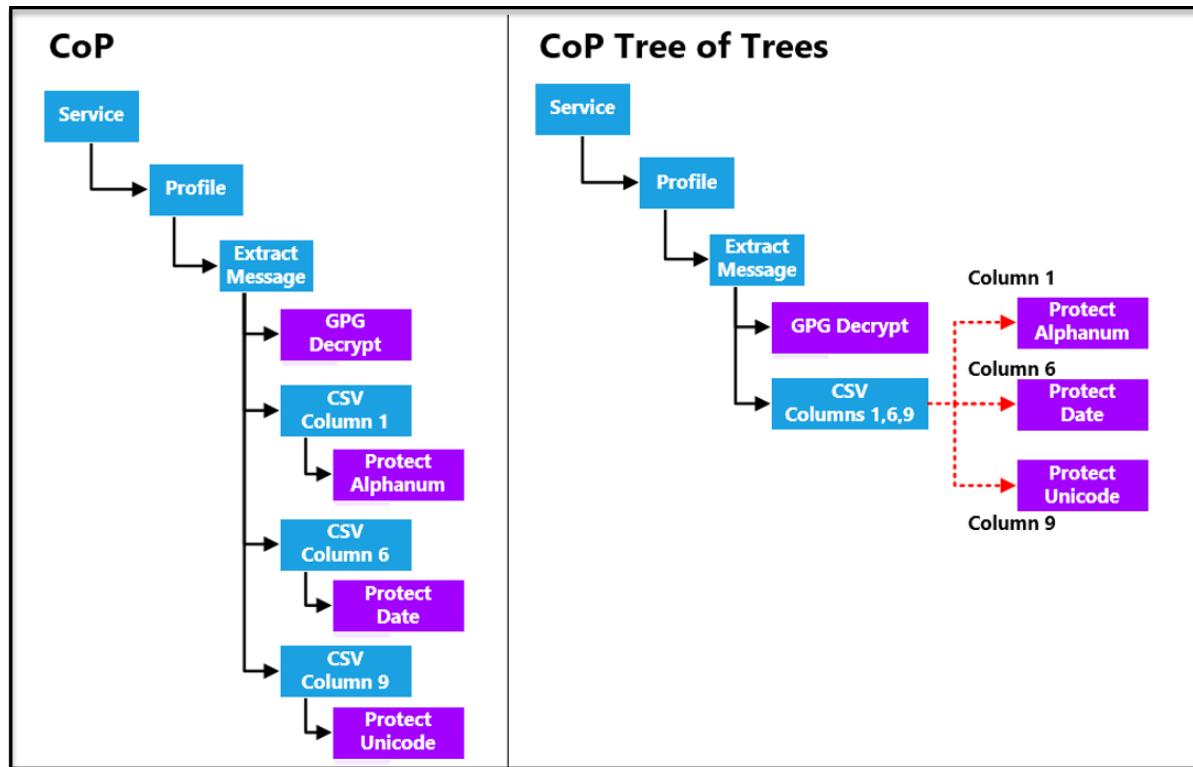


Figure 4-6: CoP vs CoP Tree of Trees

In the above example, a CSV payload needs to be processed as per the following requirements:

- Column 1 needs to be protected using an Alphanumeric data element
- Column 6 needs to be protected using a Date data element
- Column 9 needs to be protected using a Unicode data element

The traditional RuleSet strategy involved extracting or parsing the same CSV payload three times, namely once for each column that needs protection using different data elements, as shown on the left side. In contrast, a ToT-enabled RuleSet requires extracting the CSV payload only once where values extracted from different columns can be sent down different child rules that provide different protection data elements. Consequently, the overall CSV payload processing time reduces substantially.

In this release, the Ruleset ToT feature supports the [Text](#), [Binary](#), [MS Office](#), [CSV](#), [JSON with Tree-of-Trees](#) and [XML with Tree-of-Trees](#) payloads.

4.2.5 Dynamic Configuration over Programming (CoP)

Ruleset execution can be segregated into Static CoP and Dynamic CoP. When the payload type and structure is predictable in nature and known at system configuration time, you can define Rulesets for such payloads and process the data using Static CoP.

It is assumed in [Static CoP](#) that a user who defines Rulesets is authorized and holds permission to access DSG nodes.

When organizations are divided into disparate systems or applications, where each system user needs to send custom payloads on-the-fly to the DSG nodes with very little scope for predictability, providing users with access to DSG nodes for defining

Rulesets is risky. In such situations, you can use Dynamic CoP to send rules for extraction and transformation as part of a request header along with the data for protection in request message body.

Caution:

It is recommended that when creating Rulesets for Dynamic CoP, the *Profile Reference* rule is used for data transformation instead of the *Transform* rule. The security benefits of using *Profile Reference* rule are higher than the *Transform* rule since the requests can be triggered out of the secure network perimeter of an organization.

Dynamic CoP provides the following advantages:

- Flexibility to send custom requests based on the payload at hand without prior customization to Ruleset configuration.
- Restrict or configure the allowed *actions* that users can send in the request header.

The following figure illustrates how Static CoP RuleSets are combined with Dynamic CoP RuleSets as part of a given REST API or Gateway transaction:

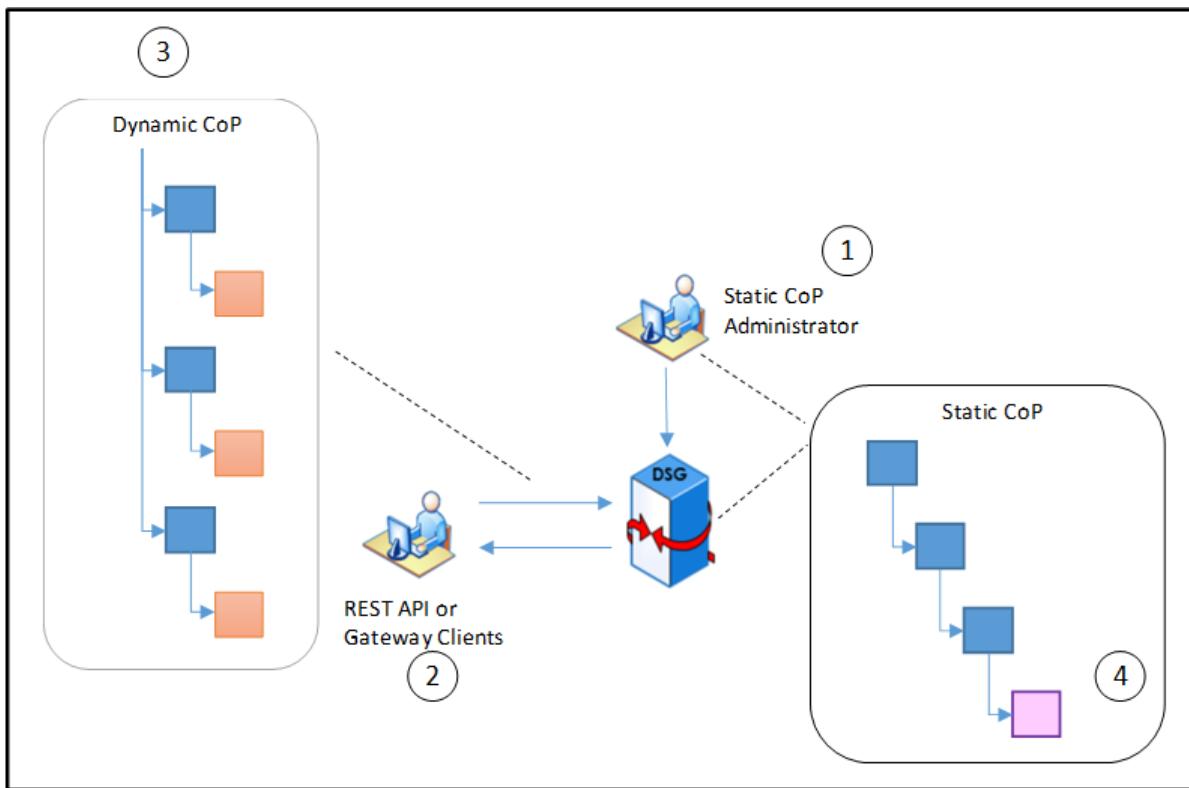


Figure 4-7: Dynamic CoP

1. The Static CoP Administrator creates the tunnel configurations and Ruleset for the Static CoP rule execution. This static rule forms the base for the Dynamic rule to follow. Based on the URI defined in both the Static CoP rule and Dynamic CoP rule, the entire Ruleset structure is executed when a request is received.
2. The REST API or gateway clients can be application developers of multiple applications in an organization who want to protect their data on-the-fly. The developers will use the Dynamic CoP concept and the allowed list of *action types* and *payloads* that they are permitted to send as request header, to create the dynamic CoP structure.
3. The *Dynamic CoP structure* provides an outline of how the request header must be constructed.
4. When the request is sent, the header hooks to the *Dynamic Injection* action type that is part of the Ruleset structure, which will be executed. The Ruleset executes successfully and protected data is sent as a response.

4.2.5.1 Dynamic CoP structure

Based on the type of Ruleset execution you want to achieve, Dynamic CoP can either be implemented with ToT or without ToT.

The following structure explains Ruleset structure when Dynamic CoP is implemented without *ToT*.

```
"rules": [
    {
        "action": {
            "payload": {
                "encoding": {
                    "type": "NO_ENCODING"
                },
                "hasToMatch": "",
                "pattern": "",
                "type": "TEXT"
            },
            "type": "EXTRACT"
        },
        "enabled": true,
        "name": "CEF Parsing",
        "rules": [
            {
                "action": {
                    "method": {
                        "dataElementName": "PTY_UNICODE",
                        "encoding": {
                            "type": "NO_ENCODING"
                        },
                        "method": "Protect",
                        "type": "PROTEGERY_DATA_PROTECTION"
                    },
                    "type": "TRANSFORM"
                },
                "enabled": true,
                "name": "Data Protection"
            }
        ]
    }
]
```

Figure 4-8: Dynamic CoP without ToT

The following structure explains Ruleset structure when Dynamic CoP is implemented with ToT.

```

"rules": [
    {
        "action": {
            "payload": {
                "encoding": {
                    "type": "NO_ENCODING"
                },
                "hasToMatch": "",
                "pattern": "src=(.*?)\\s+.*?dst=(.*?)\\sduser=(.*?)\\s+",
                "patternGroups": [
                    {
                        "group_id": 1,
                        "profileName": "helpers\\http\\helperprotectalphanumericstring"
                    },
                    {
                        "group_id": 2,
                        "profileName": "helpers\\http\\helperprotectalphanumericstring"
                    },
                    {
                        "group_id": 3,
                        "profileName": "helpers\\http\\helperprotectalphanumericstring"
                    }
                ],
                "type": "TEXT"
            },
            "type": "EXTRACT"
        },
        "enabled": true,
        "name": "CEF Parsing",
        "rules": []
    }
]

```

Figure 4-9: Dynamic CoP with ToT

Note: In the Figure, the *profileName* is the *profile reference* to the profile that the ToT structure follows. Ensure that you understand the Ruleset structure/hierarchy at the DSG node before configuring the Dynamic CoP with ToT rule.

For more information about how to create rules that process Dynamic CoP requests, refer to [Dynamic Rule](#) and [Dynamic Rule Injection](#).

4.2.5.2 Dynamic CoP Usecase

Let's review a use case where a client requests protection of certain sensitive information within a JSON document through DSG's REST API.

4.2.5.2.1 Use case implemented using Static CoP

This section provides information about the usecases implemented using static CoP.

The following image explains how the use case would be implemented if static CoP is used.

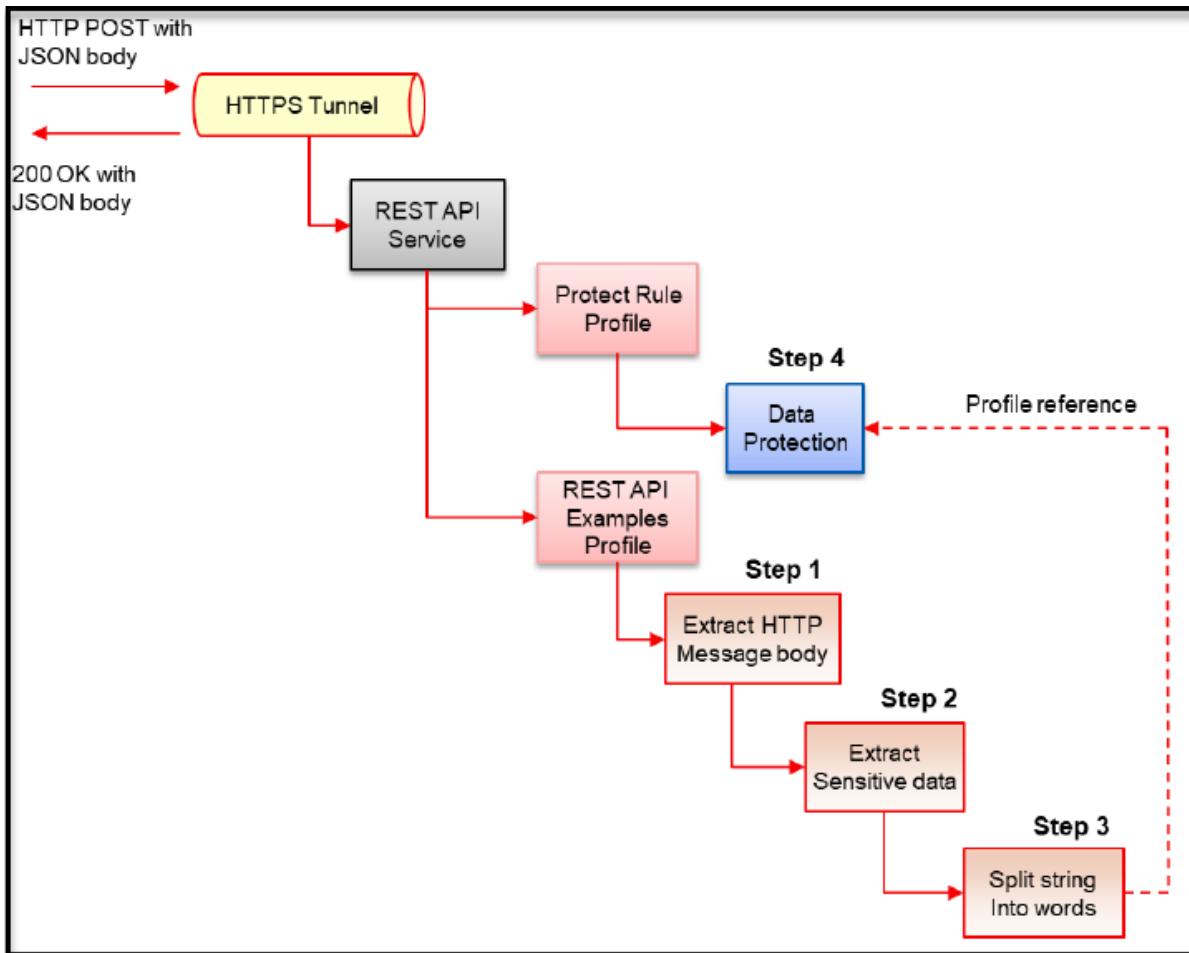


Figure 4-10: PII Usecase with Static CoP

The individual steps are described as following.

- **Step1** – This step extracts the body of the HTTP request message. The extracted body content will be the entire JSON document in our example. The extracted output of this Rule will be fed to all its children sequentially. In this example, there is only one child of this extraction rule which is step 2.
- **Step 2** – This step parses the JSON input as a text document such that a regular expression can be evaluated on it to find sensitive data within the document. This step will yield person name strings “Joe Smith” and “Alice Miller” to this children rules. In this example, there is only one child of this extraction rule which is step 3.
- **Step 3** – This step splits the extracted data from the previous rule into words. Step number 2 above yielded all person names in the document as strings and this rule in step 3 will split those strings into words such that names can be protected word by word. This will be done by running a simple REGEX on the input. Each word (e.g. “Joe”, “Smith”, “Alice” etc.) will be fed into children rule nodes of this rule one by one. In this use case, there is only one child to this rule, which is step 4.
- **Step 4** – This step does the actual data protection. Since this rule is a transformation node (a leaf node without any children), the rule will return resulting ciphertext or token to the parent.

At the end of Step 4, the RuleSet recursion stack will unwind and each branch Rule node will reverse its previous action such that the overall data can be brought back to its original format. Going back in the reverse direction, Step 4 will return tokens to Step 3 which will concatenate them together into a string. Step 2 will substitute the strings yielded back from Step 3 into the original JSON document in place of the original plaintext strings. Step 1 that was originally responsible for extracting the body of the HTTP request will now replace what has been extracted with the modified JSON document. A layer of platform logic outside of RuleSet tree execution will create an HTTP response message which will convey the modified JSON document back to the client.

4.2.5.2.2 Use case implemented using Dynamic CoP

This section provides information about the usecases implemented using dynamic CoP.

The following image explains how the use case would be implemented if dynamic CoP is used.

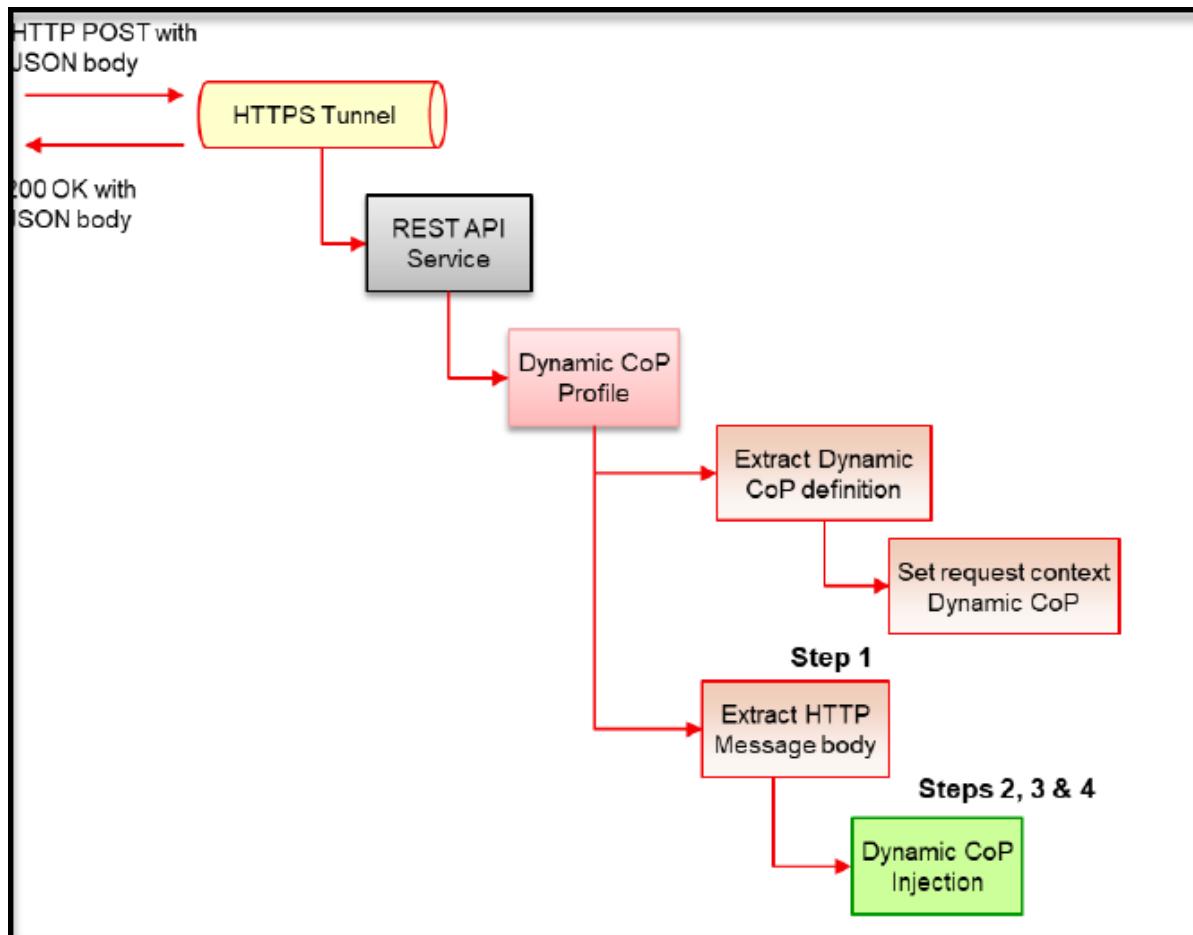


Figure 4-11: PII Usecase with Static CoP

Among the 4 steps described in [Use case implemented using Static CoP](#), steps 2 and 3 are the ones that dictate the real business logic that may change on a request by request basis. Step 1 defines extraction of HTTP request message body which is standard in any REST API request processing. Step 2 defines how sensitive data is extracted from input JSON message. Step 3 defines how a string is split into words for word-by-word protection and Step 4 defines the data protection parameters (e.g. data element name).

The logic for step 4 (data protection) can either be injected through Dynamic CoP or used through Static CoP using the profile reference concept where the protection rule is statically configured in the system and can be referenced from step 3's Dynamic CoP JSON rule. Users may choose to use statically configured protection rules and allow developers to use their profile references for an additional layer of security controls (separation of duty) and governance.

In the example, step 4's logic will be injected through Dynamic CoP to demonstrate how you can convey data element name and policy user's identity through Dynamic CoP.

4.2.5.2.2.1 Dynamic CoP Ruleset Configurations

The Dynamic CoP JSON uses the same JSON structure as the Static CoP JSON.

The only difference is that Dynamic CoP JSON is dynamically injected. To start off with our Dynamic CoP JSON, parts of the corresponding Static CoP JSON have been copied. You can create the Dynamic CoP JSON programmatically or use canned JSON template strings and substitute the variable values in it on a request-by-request basis.

The RuleSet JSON fragment for steps 2, 3 and 4 is shown in the following figure. This JSON will be delivered as-is in an HTTP header (configured as “X-Protegrity-DCoP-Rules” in our example). DSG will extract this configured header name and inject its value during run-time RuleSet tree execution.

```

1  [{"text": "Extract sensitive data ",  

2   "enabled": true,  

3   "action": {  

4     "payload": {  

5       "encoding": {  

6         "type": "NO_ENCODING"  

7       },  

8       "hasToMatch": "",  

9       "pattern": "\\"name\\\"\\s*:\\s*(.*?)\\\"",  

10      "patternGroups": [{  

11        "group_id": 1  

12      }],  

13      "type": "TEXT"  

14    },  

15    "type": "EXTRACT"  

16  },  

17  {"children": [{"text": "Extract words from input string",  

18   "enabled": true,  

19   "action": {  

20     "payload": {  

21       "encoding": {  

22         "type": "NO_ENCODING"  

23       },  

24       "hasToMatch": "",  

25       "pattern": "\\\\w+",  

26       "patternGroups": [{  

27         "group_id": 0  

28      }],  

29      "type": "TEXT"  

30    },  

31    "type": "EXTRACT"  

32  },  

33  {"children": [{"text": "Data Protection",  

34   "enabled": true,  

35   "action": {  

36     "method": {  

37       "dataElementName": "PTY_UNICODE",  

38       "encoding": {  

39         "type": "NO_ENCODING"  

40       },  

41       "method": "Protect",  

42       "type": "PROTEGILITY_DATA_PROTECTION"  

43     },  

44     "type": "TRANSFORM"  

45   }]  

46 },  

47 ]}  

48 }]  

49 }]  

50 }]  

51 }]

```

Step 2 rule – Extract "name" field in input JSON using REGEX.

Step 3 rule -- Split string into words

Data element name and user id can be fed through Dynamic CoP if needed.

Step 4 rule – Protect data

Figure 4-12: Dynamic CoP JSON Rules

The first following figure shows the skeletal Static CoP RuleSet configuration in ESA WebUI for enabling Dynamic CoP. The second following figure shows how the Dynamic CoP rules are conveyed to DSG in an HTTP header field and the JSON response output in the Postman tool. Note that the JSON response output is the same in both our Static and Dynamic CoP examples.

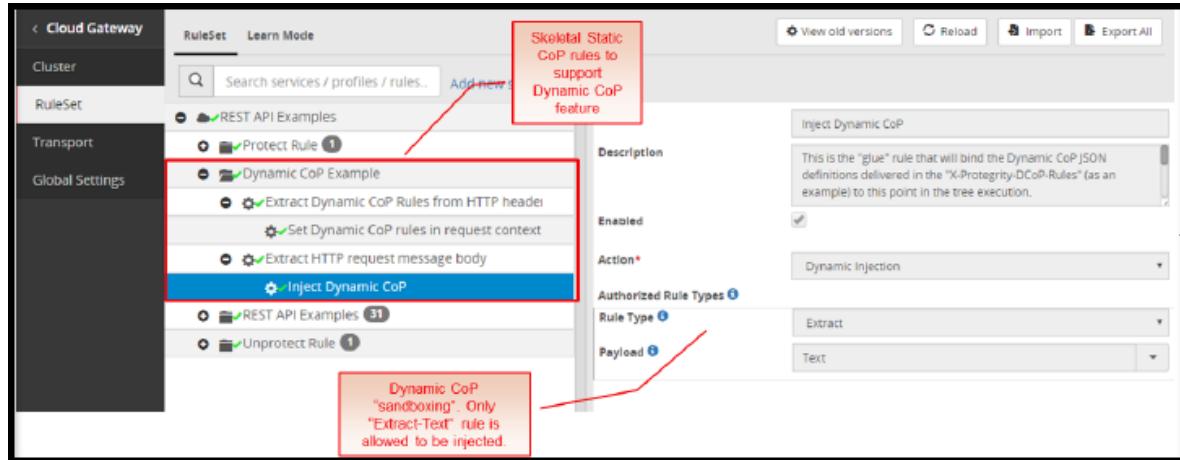


Figure 4-13: CoP Ruleset configuration Step 1 to support Dynamic CoP (step 2, 3, and 4)

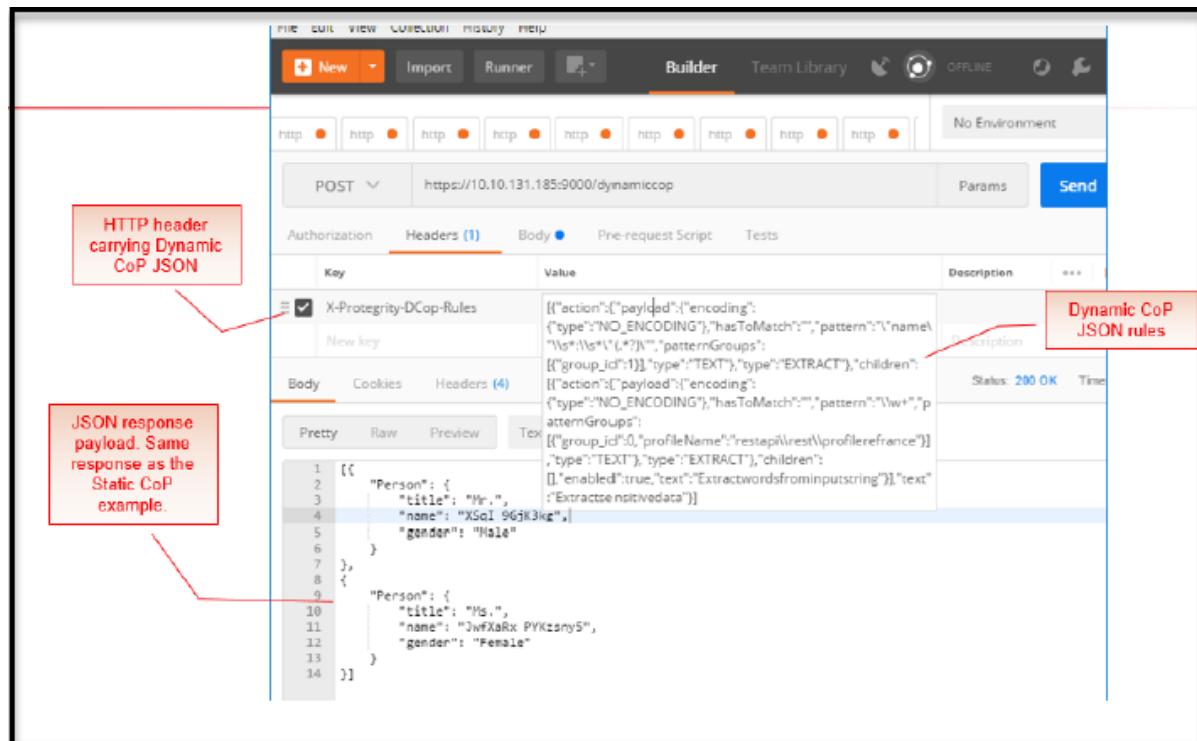


Figure 4-14: Dynamic CoP request and response header in Chrome Postman tool

4.2.6 Ruleset Execution Engine

Rulesets are executed with the Ruleset engine that is built into the gateway. The Ruleset Engine is responsible for cascaded execution of the Ruleset. The behaviors of Rules objects range from data processing (Extract and Transform) to controlling the execution flow of rule tree (Exit) to supplementary activities such as logging (Log).

The Ruleset engine will recursively traverse the Ruleset node by node. For example, Extract nodes will extract data that will be transformed with a Transform rule node. Following this, the recursion stack is rolled up and the reverse process happens where data is encoded and packaged back to its original format and sent to the intended recipient.

4.2.7 Ruleset and Ruleset Execution Example

You can follow an example of the Ruleset, the Ruleset structure, and the Ruleset execution with the following example. This example is started with an HTTP POST with an XML payload of a person's information. The Ruleset is a hierarchy of 3 Extract nodes with the Transform rule as the end leaf node.

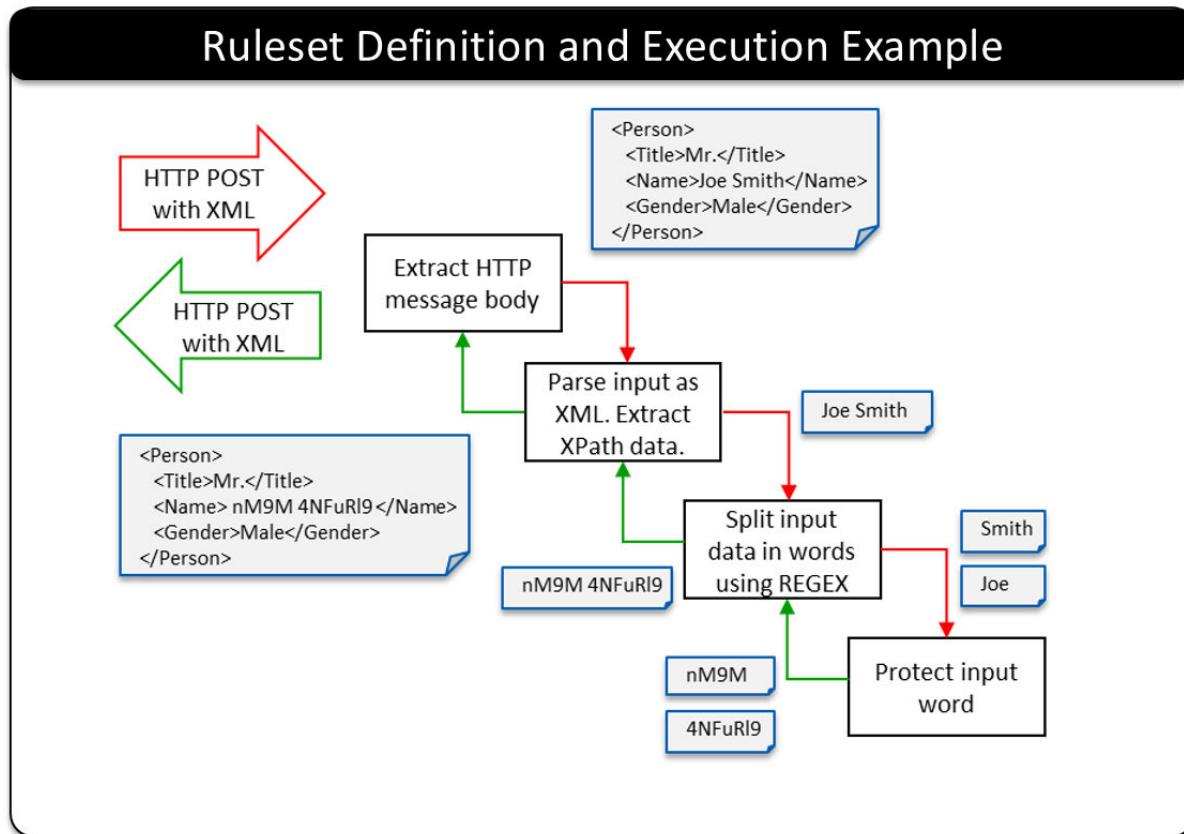


Figure 4-15: Ruleset Definition and Execution Example

- **Extract Rule:** The Extract Rule extracts the XML document from the message body.
- **Extract Rule:** A second Extract Rule will take the XML document and parse the data that is to be transformed – the person's name. This is done by using XPath.
- **Extract Rule:** A third Extract Rule will split out the name into individual words – in this example, the first and the last name. This is done by using REGEX.
- **Transform Rule:** The Transform Rule will take each word and apply an action. In this example the first name is protected and the last name is protected.

The next set of rules will perform operations in the reverse and prepare the contents to go back to the sender. The same Extraction rules would perform reverse processing as the recursion unwinds.

- **Extract Rule:** On the return trip, an Extract Rule is used to combine the protected first and last name into a single string – Name.
- **Extract Rule:** This rule will place the Name back into the XML document.
- **Extract Rule:** The final Extract rule will place the XML document back into the message body to be sent back to the sender with the name protected.

4.3 Codebook Re-shuffling

This section provides an overview for Codebook Re-shuffling.

Codebook Re-shuffling is a feature that provides an organization the ability to share protected data outside of its tokenization domain to meet data privacy, analysis, and regulatory requirements. A tokenization domain or a token domain can be defined as a business unit, a geographical location, or a subsidiary organization where protected data is stored. The data protected by enabling Codebook Re-shuffling cannot be unprotected outside the tokenization domain.

Codebook Re-shuffling provides support for the following tokenization data elements:

- Alpha (a-z, A-Z)
- Alpha-Numeric (0-9, a-z, A-Z)
- Binary
- Credit Card (0-9)
- Date (YYYY-MM-DD)
- Date (DD/MM/YYYY)
- Date (MM/DD/YYYY)
- DateTime Date (YYYY-MM-DD HH:MM:SS MMM)
- Decimal (numeric with decimal point and sign)
- Email
- Integer
- Lower ASCII (Lower part of ASCII table)
- Numeric (0-9)
- Printable
- Uppercase Alpha (A-Z)
- Uppercase Alpha-Numeric (0-9, A-Z)
- Unicode
- Unicode Base64
- Unicode Gen2

For more information about the type of tokenization data elements supported by Protegility, refer to section *Protegility Tokenization* in the [Protection Methods Reference Guide 9.1.0.0](#).

Caution: Ensure that you do not unprotect historically protected data using any token element as it causes data corruption if the *shufflecodebooks* parameter in the *pepserver.cfg* file is set to *yes*.

For example, if you have protected sensitive data using the *Credit Card* token in earlier releases of DSG, where Codebook Re-shuffling is not supported for any tokens, and upgrade to the latest version of the DSG, then unprotecting the sensitive data using the same *Credit Card* token causes data corruption if the parameters in the *pepserver.cfg* file are configured for Codebook Re-shuffling.

Note: As the Codebook Re-shuffling feature is an advanced functionality, you must contact the Protegility Professional Services team for more information about its usage.

Codebook Re-shuffling is enabled on the DSG for all the supported tokenization data elements to generate unique tokens for protected values across the tokenization domains.

The following generic example will help you to understand more about the functionality of Codebook Re-shuffling.

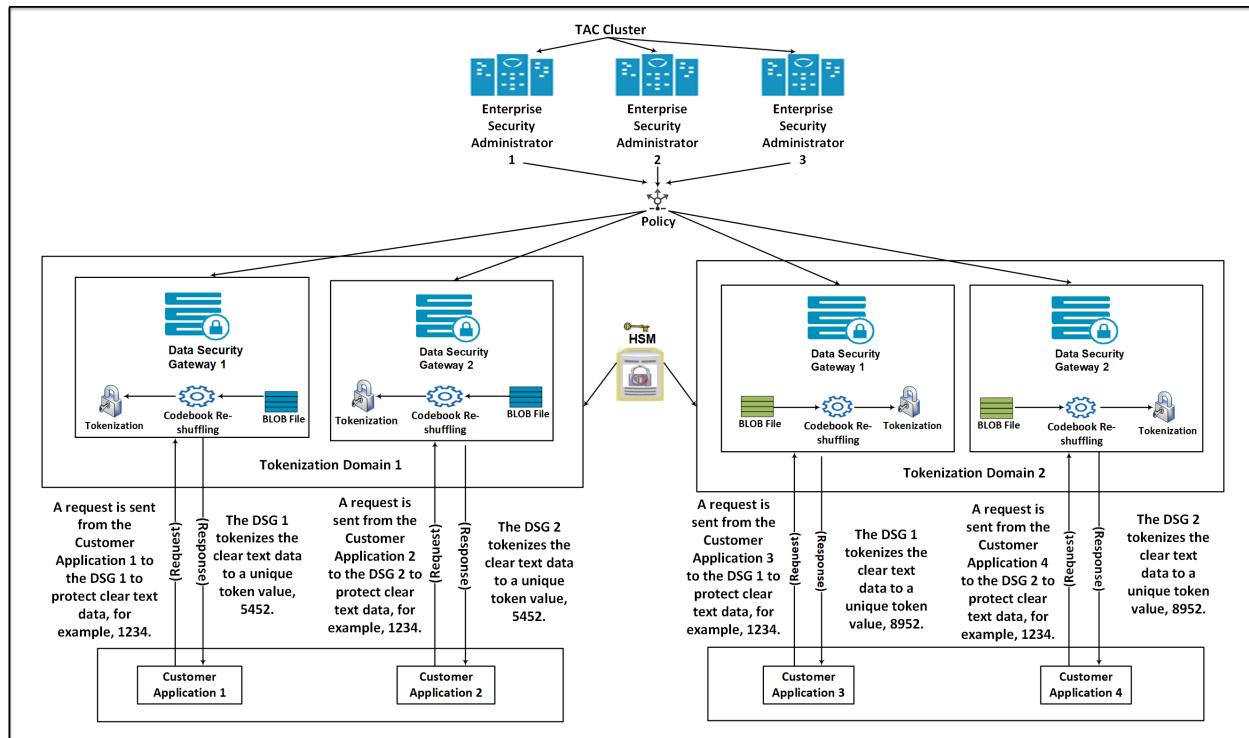


Figure 4-16: Codebook Re-Shuffling

1. Consider a scenario where an organization has two tokenization domains, Tokenization Domain 1 and Tokenization Domain 2, which are distributed in two different tokenization domains. The Tokenization Domain 1 contains an ESA connected to multiple DSG nodes. The Tokenization Domain 2 contains another ESA connected to multiple DSG nodes. There is a TAC (Trusted Appliances Cluster) setup between multiple ESA and DSG nodes.
2. On the ESA, create a tokenization data element.

Note: For more information about creating a tokenization data element, refer to section *Working with Data Elements* in the [Protegility Policy Management Guide 9.1.0.5](#).

Codebooks are generated on the ESA when a tokenization data element is created.

3. Add the newly created tokenization data element to a policy.

Note: For more information about adding a data element to a policy, refer to section *Creating and Deploying Policies* in the [Protegility Policy Management Guide 9.1.0.5](#).

4. Create a Binary Large Object (BLOB) file on each DSG node using the BLOB creation utility (BCU) that contains random bytes. The BLOB file will be encrypted automatically using an AES encryption key fetched from the HSM.
5. Deploy the policy created on the master ESA to the DSG nodes in Token Domain 1 and Token Domain 2. The DSG nodes download the policy information from the ESA. After the policy is deployed on the DSG, if the codebook re-shuffling parameter is enabled, then the codebook will be shuffled again by using the BLOB file created on the DSG.

Note: For more information about deploying a policy, refer to section *Creating and Deploying Policies* in the [Protegility Policy Management Guide 9.1.0.5](#).

6. Create a Ruleset on the DSG nodes to protect the sensitive data.

After a request is sent from the client, the DSG processes and protects the sensitive data. It generates unique tokens for protected values across the tokenization domains.

The advantage offered by codebook re-shuffling is if a data analyst on the DSG node in the Token Domain 2 wants to access the sensitive data of Token Domain 1, then it cannot be accessed because it is protected by a different codebook, available only on the

Token Domain 1. The unique tokens generated for the protected data on the Token Domain 1 can be used to derive an insightful analysis for an organization without compromising on any data security compliance and regulatory norms.

4.3.1 Codebook Re-shuffling in the PEP Server

Codebook Re-shuffling in the PEP server uses a utility to create a Binary Large Object (BLOB) file that contains random bytes. The file with random bytes is encrypted using an AES encryption key from the HSM and saved to the disk.

The PEP server loads the BLOB file from the disk and decrypts it using the key from the HSM. It then re-shuffles the supported codebooks, whenever a policy is published to shared memory. Codebook Re-shuffling generates unique tokens for protected values across the tokenization domains.

Caution: Ensure that you do not unprotect historically protected data using any token element as it causes data corruption if the *shufflecodebooks* parameter in the *pepper.cfg* file is set to *yes*.

Note: After the data is protected by enabling Codebook Re-shuffling on the DSG, you must perform data security operations like protect, unprotect, and re-protect for the sensitive data by only using the Data Security Gateway (DSG) protector. Also, DSG does not support the migration of protected data, with the *shufflecodebooks* parameter in the *pepper.cfg* file is set to *yes*, from the DSG to other protectors. An attempt to migrate the protected data and unprotecting it may cause data corruption.

Note: The Codebook Re-shuffling feature is tested and supported for the Safenet Luna 7.4 HSM device. The procedure provided in this section is for the Safenet Luna 7.4 HSM device.

► To enable the re-shuffling of codebooks in the PEP server:

1. Download the HSM library files. In this procedure, it is assumed that the HSM files are added to the */opt/protegility/hsm* directory.

Note: The HSM directory is not created on the DSG by default after DSG installation. Ensure that you use the following commands to create the HSM directory.

```
mkdir /opt/protegility/hsm
```

2. Ensure that the required ownership and permissions are set for the HSM library files in the */opt/protegility/hsm* directory by running the following commands.

```
chown -R service_admin:service_admin /opt/protegility/hsm
chmod -R 744 /opt/protegility/hsm
```

3. Ensure that the HSM library configuration files are available on the DSG.
4. Create a soft link to link the HSM shared library file using the following commands.

```
cd /opt/protegility/defiance_dps/data
su -s /bin/sh service_admin -c "ln -s /opt/protegility/hsm/<HSM shared library file>
pkcs11.pml"
```

5. Create the environment settings file using the following commands.

```
echo "export <Variable Name>=<HSM Configuration file path> >> /opt/protegility/defiance_dps/bin/dps.env
chown service_admin:service_admin /opt/protegility/defiance_dps/bin/dps.env
chmod 644 /opt/protegility/defiance_dps/bin/dps.env
```



- Export the *HSM Configuration file* parameter in the current session using the following command.

```
source ./bin/dps.env
```

- Create the AES encryption key in the HSM using the following commands.

```
cd /opt/protegility/defiance_dps/data
./bin/bcu -lib ./pkcs11.plm -op createkey -label <labelname> -slot <slotnumber> -userpin <SOME_USER_PIN>
```

- Create a BLOB file using the BLOB creation utility (BCU), encrypt the BLOB file with the created encryption key, and save the BLOB file to the disk using the following command.

```
./bin/bcu -lib ./pkcs11.plm -op createblob -label <labelname> -slot <slotnumber> -size <size> -userpin <password> -file random.dat
```

- Create the credentials file for the PEP server to connect to the HSM using the following command.

```
./bin/bcu -op savepin -userpin <password> -file userpin.bin
```

- Ensure that the required ownership and permissions are set for *random.dat* and *userpin.bin* files by using the following command.

```
chown service_admin:service_admin userpin.bin random.dat
```

Caution: Ensure that you backup the BLOB file and the *user pin* file and save it on your local machine. The BLOB file and the *user pin* files must not be saved on the DSG.

- Add the *shufflecodebooks* configuration parameter and the path to the file containing the random bytes in the *pepper.cfg* configuration file using the following format:

```
# shuffle token codebooks after they are downloaded.
# yes, no. default no.
shufflecodebooks = yes

# Path to the file that contains the random bytes for shuffling codebooks.
randomfile = ./random.dat
```

Note: The *shufflecodebooks* configuration parameter should be added in the *Policy Management* section of the *pepper.cfg* file.

- Add the following snippet for the *PKCS#11 configuration* in the *pepper.cfg* configuration file. After adding the snippet, the user must modify the required path to the *PKCS#11* provider library, the slot number to be used on the HSM, and the required path to the *userpin.bin* file in the *pepper.cfg* configuration file.

```
# -----
# PKCS#11 configuration
# Values in this section is only used
# when shufflecodebooks = yes
# -----
[pkcs11]

# The path to the PKCS#11 provider library.
provider_library = ./pkcs11.plm

# The slot number to use on the HSM.
slot = 1 /Enter the slot number used at the time of the creation of the key/

# The scrambled user pin file.
userpin = ./userpin.bin
```

Note:

Ensure that the slot number added in [step 8](#) of this procedure and the slot number added in [step 12](#) are the same.

13. On the DSG Web UI, navigate to **System > Services** to restart the PEP server.

4.3.2 Re-protecting the Existing BLOB

This section describes the steps to reprotect the existing BLOB file with a new key. The user can encrypt the BLOB file by using the steps mentioned in the [Codebook Re-shuffling in the PEP Server](#) section. These steps will allow the user to encrypt the BLOB using a key label of an AES key. After performing these operations, if the user wants to reprotect the BLOB with another key, then the user should create a new key and use the new key label to encrypt the BLOB. The reprotect operation provided will only reprotect the existing BLOB with a new key label and will not change the content in the BLOB.

► **To reprotect the existing BLOB:**

1. Create the new AES encryption key in the HSM using the following commands.

```
./bcu -lib <safenet lib so file path> -slot <slotId> -userpin <userpin>
-op createkey -label <new_labelname>
```

2. Run the following command to reprotect the existing BLOB.

```
./bcu -lib <safenet lib so file path> -slot <slotId> -userpin <userpin>
-op reprotectblob -label <new_labelname> -file <blob filename>
```

Note: The BCU utility will not perform the reprotect operation of the BLOB, if the keys are used from different HSMs or from different slots or partitions.

3. Ensure that the required ownership and permission are set for the random.dat file by using the following commands.

```
chown service_admin:service_admin random.dat
chmod 640 random.dat
```

4. To have the changes made in the above steps reflected, login to the DSG Web UI and navigate to **System > Services** to restart the PEP server. You can view the success or failure logs of the re-shuffling process on the PepServerLog screen on the DSG Web UI. To view the PepServerLog screen, navigate to **Logs > PepServer**.

On the PepServerLog screen, ensure that you have set the Log level to **ALL** in the pepserver.cfg file.

```
# -----
# Logging configuration,
# Write application log to file as trace
# -----
[logging]
# Logging level: OFF - No logging, SEVERE, WARNING, INFO, CONFIG, ALL
level = ALL
```

The following figure shows the PepServerLog after the BLOB file is encrypted with new key.

```
Protegrity PEP Server
Version : [REDACTED]
Copyright (c) 2013 Protegrity Corporation. All Rights Reserved.
  (INFO) Application starting, pid=11441
  (INFO) Starting Protegrity PEP Server
  (INFO) Version:
  (INFO) Platform: Linux_x64
  (INFO) Hostname: [REDACTED]
  (INFO) OS User: service_admin
  (INFO) IP Address:
  (CONFIG) Policy will be downloaded from https://[REDACTED]
  (CONFIG) Policy refresh interval: 79 seconds
  (CONFIG) Codebooks downloaded from ESA will be shuffled
  (CONFIG) Using HSM library: /opt/protegrity/hsm/external/libCryptoki2_64.so, slot: 0
  (CONFIG) Using random file: ./random.dat
  (CONFIG) Decrypting BLOB using key label: <new_labelname>
  (INFO) Key handler loaded: internal
  (CONFIG) Communication id: 0
  (CONFIG) Semaphore resources: 4000
```

Chapter 5

Deployment Scenarios

The Protegrity Gateway Technology has the flexibility to be deployed on premise, on a private cloud, on a public cloud or any hybrid combination if the necessary network communication is available to its consumers.

The physical deployment approach is based on your business and security requirements for your use cases and organization.

For example, data security may require an on-premise deployment to keep the key material within physical geography or corporate logical borders. These borders mark the security perimeter, or domain, that unprotected sensitive data should not cross.

Security domain borders may also be subject to the type of sensitive data. Functioning as a demarcation point, the gateway can protect and unprotect sensitive data crossing these borders. As such, a business data security team will require the gateway to be deployed within what they consider secured domain for the subject sensitive data.

The following diagrams depict different deployment scenarios.

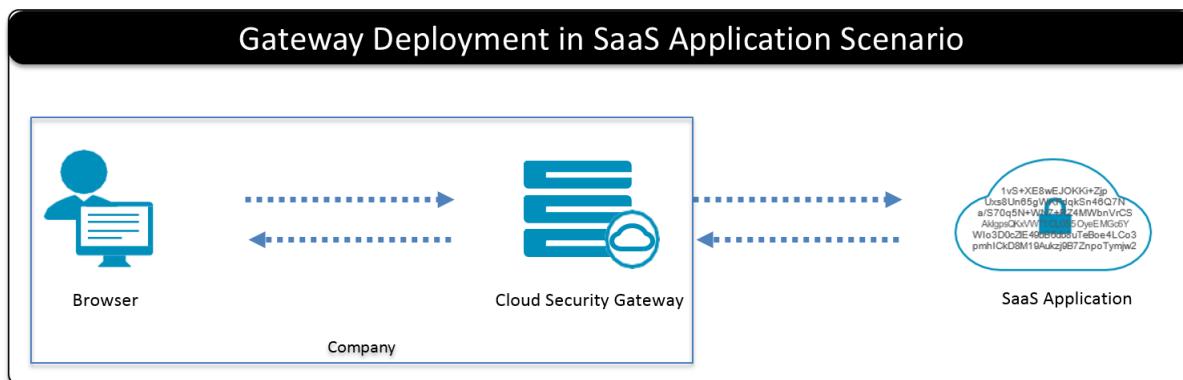


Figure 5-1: Gateway Deployment in SaaS Application Scenario

When protecting SaaS Applications, the Cloud Security Gateway is deployed on premise of the company that is using the SaaS application as a business system.

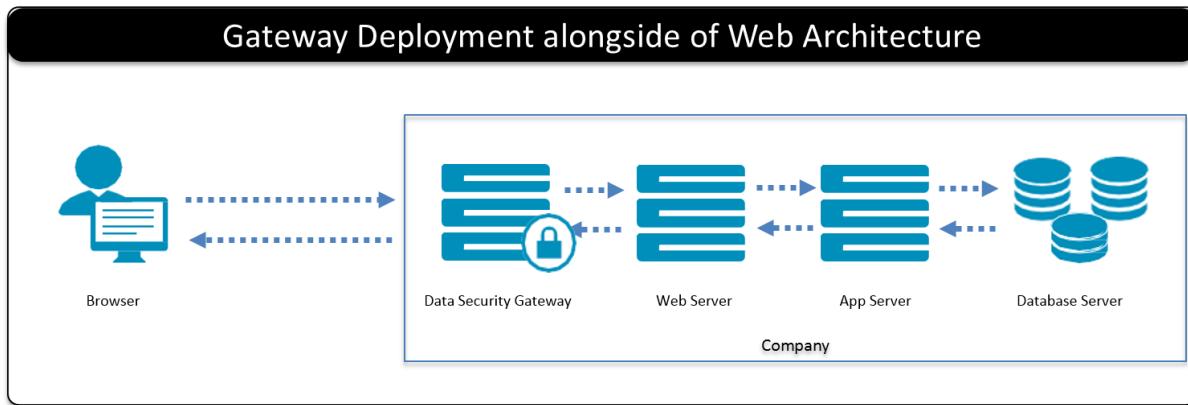


Figure 5-2: Gateway Deployment

When protecting Web Applications that are deployed on premise, the Data Security Gateway is deployed on premise of the company that is hosting the web application infrastructure.

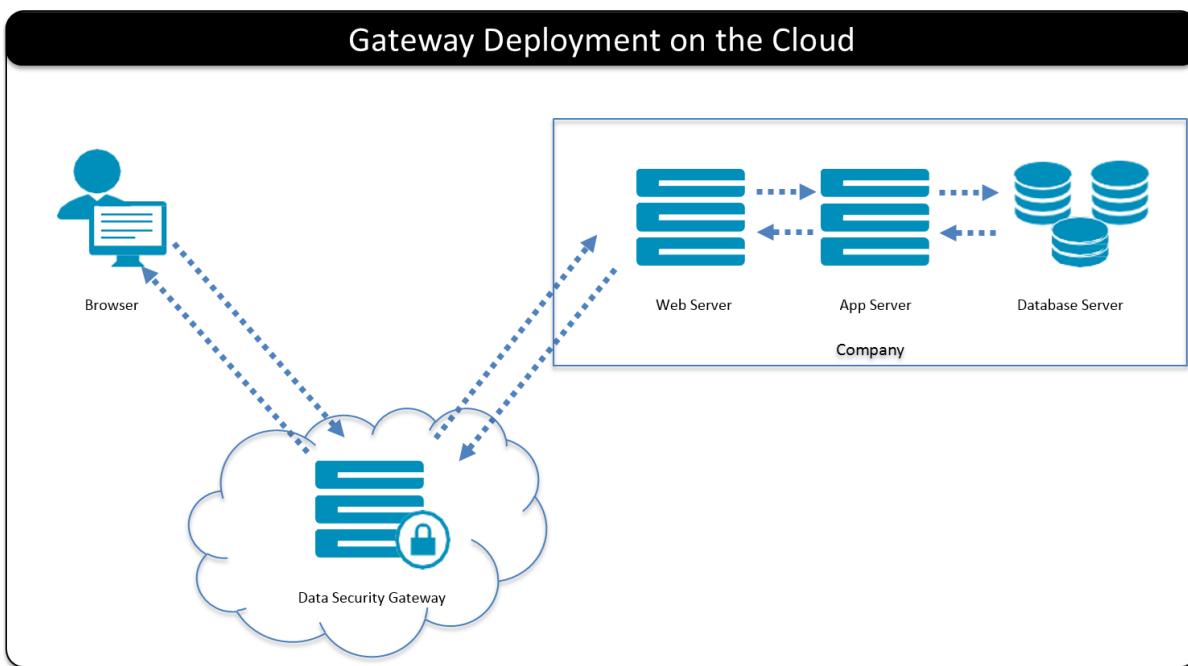


Figure 5-3: Gateway Deployment on Cloud

Companies have the option of deploying the Data Security Gateway or the Cloud Security Gateway on the cloud (private or public).

Chapter 6

Protegility Methodology

[6.1 Data Governance](#)

[6.2 Discovery](#)

[6.3 Solution Design](#)

[6.4 Product Installation](#)

[6.5 Solution Configuration](#)

[6.6 Initial Migration](#)

[6.7 Testing](#)

[6.8 Production Rollout](#)

The Protegility Methodology helps organizations implement a data security solution through a set of steps that start with data governance and ends at rolling out the implemented solution.

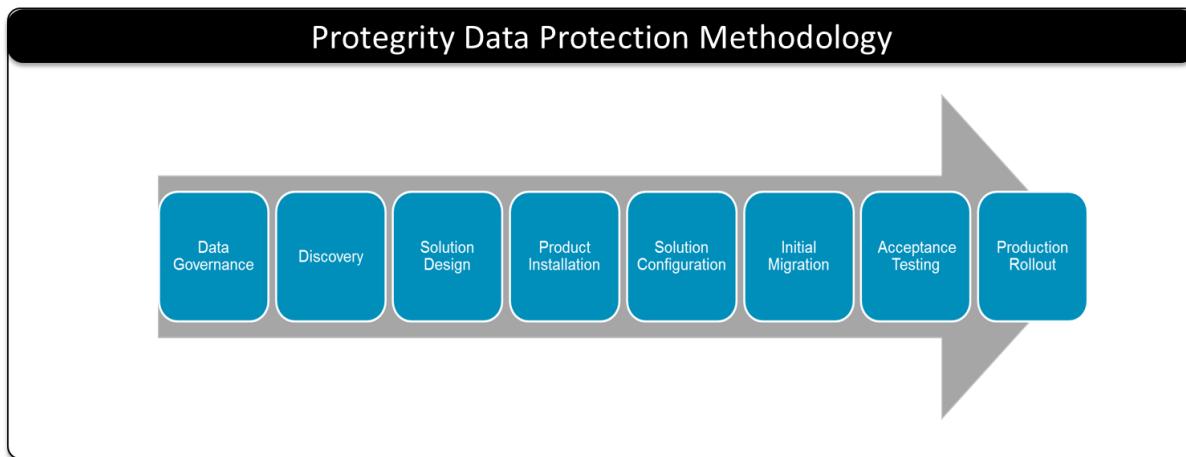


Figure 6-1: Protegility Data Protection Methodology

6.1 Data Governance

Corporate Data Governance, often based on a board level directive, will specify the data that is sensitive to an organization. The source of these data elements may come from regulatory requirements or from internal corporate security goals that go beyond standard compliance. These are the data elements that will be the focus of designing and delivering a data security solution.

6.2 Discovery

During the Discovery step, Protegility Solution Architects will collaborate with the customer corporate IT and Corporate Security stakeholders to identify the location and use of the sensitive data that has been identified by Data Governance.

A Discovery document is created that contains the data flows, technologies used (databases, applications, etc.), performance, SLA requirements, and who is authorized to view protected sensitive data in the clear.

6.3 Solution Design

Based on the results of the Discovery Step, Solution Architects will work with the customer Architecture stakeholders to design and document a data security solution that will meet the requirements from Data Governance.

This step involves methodically tracing through the Discover document, following the path of sensitive data as it flows through different technologies. The goal is to deliver end to end data security from the point of entry or creation through business processes, and ultimately until the data is archived or deleted.

At different points during this step, prototyping may be used to assess the impact of a solution over another.

The data security solution is recorded in a Solution Design document.

Protegility Data Security Solutions have the goal of delivering security to match the risk tolerance of the organization while recognizing the trade-off between security and usability.

6.4 Product Installation

The Solution Design document will identify the list of Protegility products that will be used to satisfy the customer data security requirements. These products need to be installed on the target environments.

Installation step also involves basic settings and verification of connectivity among the designed solution product components.

6.5 Solution Configuration

The Protegility platform has the flexibility to protect whatever data your organization deems sensitive and to use the most appropriate protection method. Configuring the solution means that data security policies will be created and deployed to the Protegility protectors. The policies will identify the data that needs to be protected, how that data is to be protected and who should have access to that data. These policies are deployed to all Protegility protection agent and will guide protectors on all data security operations.

In addition to the data security policy, the protectors are configured to bind the data protection operations to a target layer, system or environment. The Data Security Gateway (DSG) is integrated at the network level and therefore it is likely that the configuration step will also involve network firewall, load balancer, and IDP configuration or integration. Specific Gateway Rulesets for the designed solution will also be identified and set as part of this step.

6.6 Initial Migration

With all data security solutions where sensitive data is being changed – protected, all existing data will need to be protect as well. This process is known as Initial Migration. Initial migration is applied to replace all the sensitive data that already exists in the system unprotected, with its protected. This step exists to avoid having unprotected and protected data mixed together.

6.7 Testing

Data Security Solution add security functions that will protect and unprotect sensitive data. These security operations may be constrained to certain individuals or processes. The step in the Protegility Methodology will require the testing of the data security solution before rolling the solution out.

The focus of this methodology step is to ensure that the data is protected, when it should be protected, when it should be unprotected, and that business systems continue to function as usual all controlled by the data security policy.

6.8 Production Rollout

The final step is to roll the solution out and make it available for users.

Chapter 7

Planning for Gateway Installation

[7.1 Planning Overview](#)

[7.2 Minimum Hardware Requirements](#)

[7.3 ESA](#)

[7.4 Forwarding Logs in DSG](#)

[7.5 LDAP and SSO Configurations](#)

[7.6 Mapping of Sensitive Data Primitives](#)

[7.7 Network Planning](#)

[7.8 HTTP URL Rewriting](#)

[7.9 Clustering and Load Balancing](#)

[7.10 SSL Certificates](#)

This section provides information about prerequisites that must be met before DSG installation can be started.

7.1 Planning Overview

This section can be used as a guide and a checklist for what needs to be considered before the gateway is installed.

This document has many examples of technical concepts and activities like the ones described in this section that are part of the gateway using and configuring the gateway. As a way of facilitating the explanation of these concepts and activities, a fictitious organization called Biloxi Corp is used. The Biloxi Corp has purchased a SaaS called ffcrm.com. The Protegility gateway is used to protect Biloxi data that is stored in ffcrm.com.

7.2 Minimum Hardware Requirements

The performance of the gateway nodes is primarily dependent on the capabilities of the hardware they are installed on.

While optimal hardware server specifications are dependent on individual product usage environments, the minimum hardware specifications recommended lower end for production environments are as follows:

- CPU: 4 Cores
- Disk Size: 320 GB
- RAM: 16 GB
- Network Interfaces: 2

Note: The hardware configuration required might vary based on the actual usage or amount of data and logs expected.

The gateway software appliances are certified on the following server platforms.

Vendor	Platform Name	Details	Comments
IBM	X3250	IBM x3250 M2 INTEL XEON SATA Hard-Drive	
	X3550	IBM x3550 M2, M3, M4 INTEL XEON SATA Hard-Drive	
HP	HP Proliant DL 360 G5	HP Smart-Array 400i HP Smart Array P420i Controller	According to HP documentation, the following Smart-Array controllers are compatible: 5300, 5i, 532, 5312, 641, 642, 6400, 6400 EM, 6i, P600, P800, P400, P400i, E200i, E200, E500, P700M,
	HP Proliant DL385p G8		
DELL	PowerEdge R310	PERC H700 SATA Hard-Drive	
	PowerEdge R620	Disk Controller: PERC H310 Mini (embedded)	
VMWare	ESX/ESXi/ Server Hypervisor		ALL VMWare platforms.
Citrix XenServer	XenServer 5.6, 6.1 Hypervisor		Both Full-Hardware-Virtualization (HVM) and Paravirtualization (PVM) supported. XenTools can be installed as well.
Open Source Xen	XenSource Hypervisor		Both Full-Hardware-Virtualization (HVM) and Paravirtualization (PVM) supported.
Microsoft	Hyper-V Hypervisor		Using "Legacy Network Adapter" (Hyper-V configuration).
Red Hat	KVM Hypervisor		

Figure 7-1: Server Platforms-Gateway Software Appliances certified on

7.3 ESA

As with all Protegility protectors, gateway instances are centrally managed and controlled from the ESA. As a prerequisite to gateway installation, a working instance of the ESA is required.

Note: For information about the ESA version supported by this release of the Data Security Gateway (DSG), refer to the [Data Security Gateway v3.1.0.4 Release Notes](#).

ESA is the centrally managed component that consists of the policy related data, data store, key material, and the DSG configurations, such as, *Certificates*, *Rulesets*, *Tunnels*, *Global Settings*, and some additional configurations in the *gateway.json* file. As per design, the ESA is responsible for pushing the DSG configuration to all the DSG nodes in a cluster.

Caution: If you create any configuration on a DSG node and the deploy operation is performed on the ESA, then the configuration on the DSG node will be overwritten by the configuration on the ESA and you will lose all the configuration on the DSG node. Thus, it is recommended that if you are creating any DSG configuration, you must create it on the ESA as the same configurations will be pushed to all the DSG nodes in the cluster. This ensures that the configurations available on all the DSG nodes in a cluster are the same.

Ensure that you push the DSG configurations by clicking *Deploy* or *Deploy to Node Groups* from the ESA Web UI. You can click the *Deploy* or *Deploy to Node Groups* options from the *Cluster* and *Ruleset* screens on the ESA Web UI.

Note: Clicking the *Deploy* or *Deploy to Node Groups* options from either of these screens on the ESA Web UI ensures that all the DSG configurations are pushed from the ESA to the DSG nodes in a cluster.

7.4 Forwarding Logs in DSG

The log management mechanism for Protegility products forwards the logs to the Audit Store on the ESA.



The following services forwards the logs to the audit store:

- **td-agent** : It forwards the appliance logs to the Audit Store on the ESA.
- **Log Forwarder**: It forwards the data security operations-related logs, such as, protect, unprotect, and reprotect and the PEP server logs to the Audit Store on the ESA.

Caution: After the ESA 9.1.0.3 installation, ensure that the **Analytics** is initialized on the ESA. The initialization of the **Analytics** displays the information from the Audit Store.

For more information about initializing the **Analytics**, refer to the section *Initializing the Audit Store Cluster on the ESA* in the *Protegility Installation Guide 9.1.0.5*

For more information about configuring the DSG to forward appliance logs to the ESA, refer to the section [Forwarding Appliance Logs to the Audit Store](#).

For more information about configuring *Log Forwarder* to forward the audit logs, refer to the section [Forwarding Audit Logs to the Audit Store](#).

For more information about auditing and logging in the DSG, refer to the section [Auditing and Logging](#).

7.5 LDAP and SSO Configurations

The DSG is dependent on the ESA for user management. The users that are part of an organization AD are configured with the ESA internal LDAP.

If your organization plans to implement SSO authentication across all the Protegility appliances, then you must enable SSO on the ESA and the DSG. The DSG depends on the ESA for user and access management and it is recommended that user management is performed on the ESA.

Before you can configure SSO with the DSG, you must complete the prerequisites on the ESA.

For more information about completing prerequisites on the ESA, refer to section *Implementing SSO on DSG* in the *Protegility Appliances Overview Guide 9.1.0.5*.

After completing the prerequisites, ensure that the following order of SSO configuration on the DSG nodes is followed.

Table 7-1: Configuring SSO on the DSG

Order of Configuration	Description	Reference
1	Enable SSO on the DSG node.	Enabling SSO on DSG
2	Configure the Web browser to add the site to trusted sites.	Configuring SPNEGO Authentication on the Web Browser
3	Login to the DSG appliance	Logging to the Appliance

7.5.1 Enabling SSO on DSG

This section provides information about enabling SSO on the DSG nodes. It involves setting the ESA FQDN and enabling the SSO option.

Before you begin

Before SSO is enabled, ensure that the following prerequisite is completed.

- Ensure that the ESA FQDN is available.



► To enable SSO on the DSG node:

1. Login to the DSG Web UI.
2. Navigate to **Settings > Users**.
3. Click the **Advanced** tab.

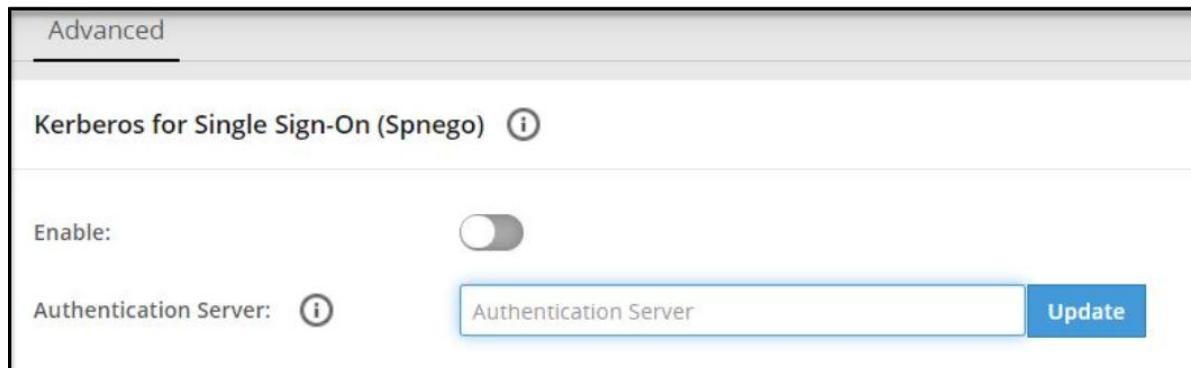


Figure 7-2: Enabling SSO Authentication on DSG

4. In the **Authentication Server** field, enter the ESA FQDN.
5. Click **Update** to save the server details.
6. Click the **Enable** toggle switch to enable the Kerberos SSO.
7. Repeat the step 1 to step 6 on all the DSG nodes in the cluster.

7.5.2 Configuring SPNEGO Authentication on the Web Browser

Before implementing Kerberos SSO for Protegility appliances, you must ensure that the Web browsers are configured to perform SPNEGO authentication. The tasks in this section describe the configurations that must be performed on the Web Browsers. The recommended Web browsers and their versions are as follows:

- Google Chrome version 123.0.6312.123 (64-bit)
- Mozilla Firefox version 124.0.2 (64-bit) or higher
- Microsoft Edge version 123.0.2420.81 (64-bit)

The following sections describe the configurations on the Web browsers.

7.5.2.1 Configuring SPNEGO Authentication on Firefox

The following steps describe the configurations on Mozilla Firefox.

► To configure on the Firefox Web browser:

1. Open Firefox on the system.
2. Enter *about:config* in the URL.
3. Type *negotiate* in the **Search** bar.
4. Double click on *network.negotiate-auth.trusted-uris* parameter.
5. Enter the FQDN of the appliance and exit the browser.

7.5.2.2 Configuring SPNEGO Authentication on Internet Explorer

The following steps describe the configurations on Internet Explorer 11.

► To configure on the Internet Explorer Web browser:

1. Open Internet Explorer on the machine
2. Navigate to **Tools > Internet options > Security**.
3. Select **Local intranet**.
4. Enter the FQDN of the appliance under sites that are included in the local intranet zone.
5. Select **Ok**.

7.5.2.3 Configuring SPNEGO Authentication on Chrome

With Google Chrome, you must set the white list servers that Chrome will negotiate with. If you are using a Windows machine to log in to the appliances, then the configurations entered in other browsers are shared with Chrome. You need not add a separate configuration.

7.5.3 Logging to the Appliance

After configuring the required SSO settings, you can login to the DSG using SSO.

► To login to the DSG using SSO:

1. Open the Web browser and enter the FQDN of the DSG in the URL.
The following screen appears.

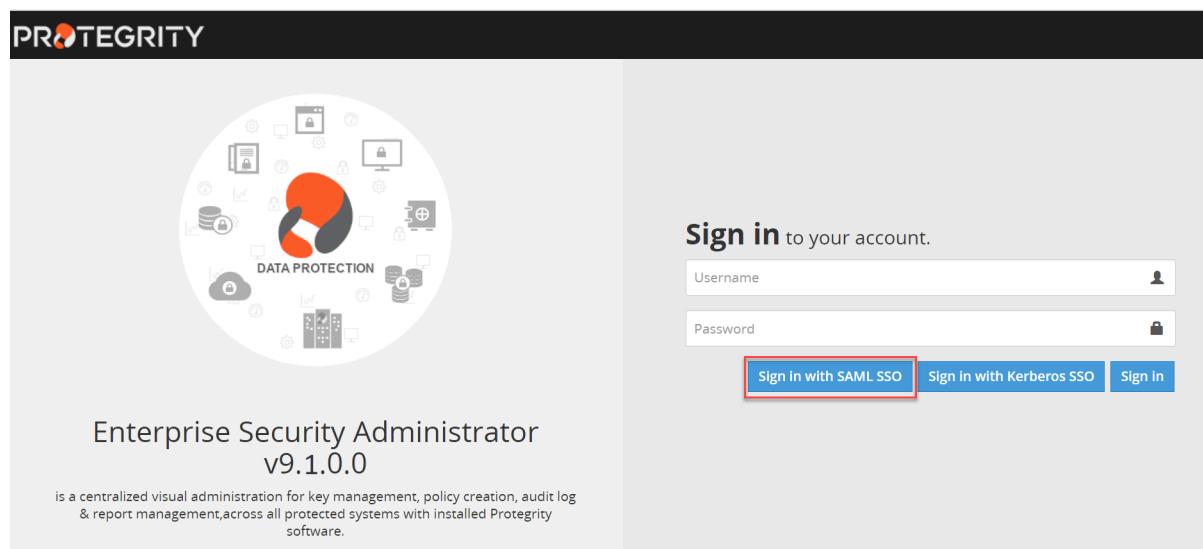


Figure 7-3: Login Screen

2. Click **Sign in with SSO**.
The Dashboard of the DSG appliance appears.

7.6 Mapping of Sensitive Data Primitives

Corporate Governance will typically identify the data that is deemed sensitive to an organization. An example of this data can be PCI DSS data such as credit cards, Personally Identifiable Data (PII) and Protected Health Information (PHI). PII can include data elements such as First name, Last Name, Social Security Numbers, E-mail Addresses, or any data element that can identify an individual.

When using the gateway to protect sensitive data, the data must be identified through techniques exposed in a CoP Profile. For example, if the requirement is to protect sensitive data in a public SaaS, the identified sensitive data will need to be mapped to the corresponding fields in web forms rendered by the SaaS. These web forms are typically part of SaaS web pages where end users input sensitive data in SaaS for adding new data or searching existing data. A later section on the gateway configuration describes how the form fields will be targeted for protection through configuration rules.

7.7 Network Planning

Connecting the gateway to a network involves address allocation and network communication routing for the service consumers. Network planning also includes gateway cluster sizing and addition of Load Balancers (LB) in front of the gateway cluster.

To protect data in a SaaS application, you gather a list of public domain and host names through which the SaaS is accessed over the Internet.

In case of internal enterprise applications, this relates to identifying networking address (IP addresses or host names) of relevant applications.

Gateway network interfaces can be divided into two categories, administrative and service. Administrative interfaces, such as Web UI and command line (SSH), are used to control and manage its configuration and monitor its state while service interfaces are used to deliver the service it is set to do. It is important that two NICs are created before you install the DSG.

For network security reasons DSG isolates the administrative interfaces from the service ones by allocating each with a separate network address. This enables physical separation when more than one physical NIC is available, otherwise logical separation is achieved by designating two different IP Addresses for admin and service use. Production implementation may strive to achieve further isolation for the service interface by separating inbound and outbound channels, in which case three IP Address will be required.

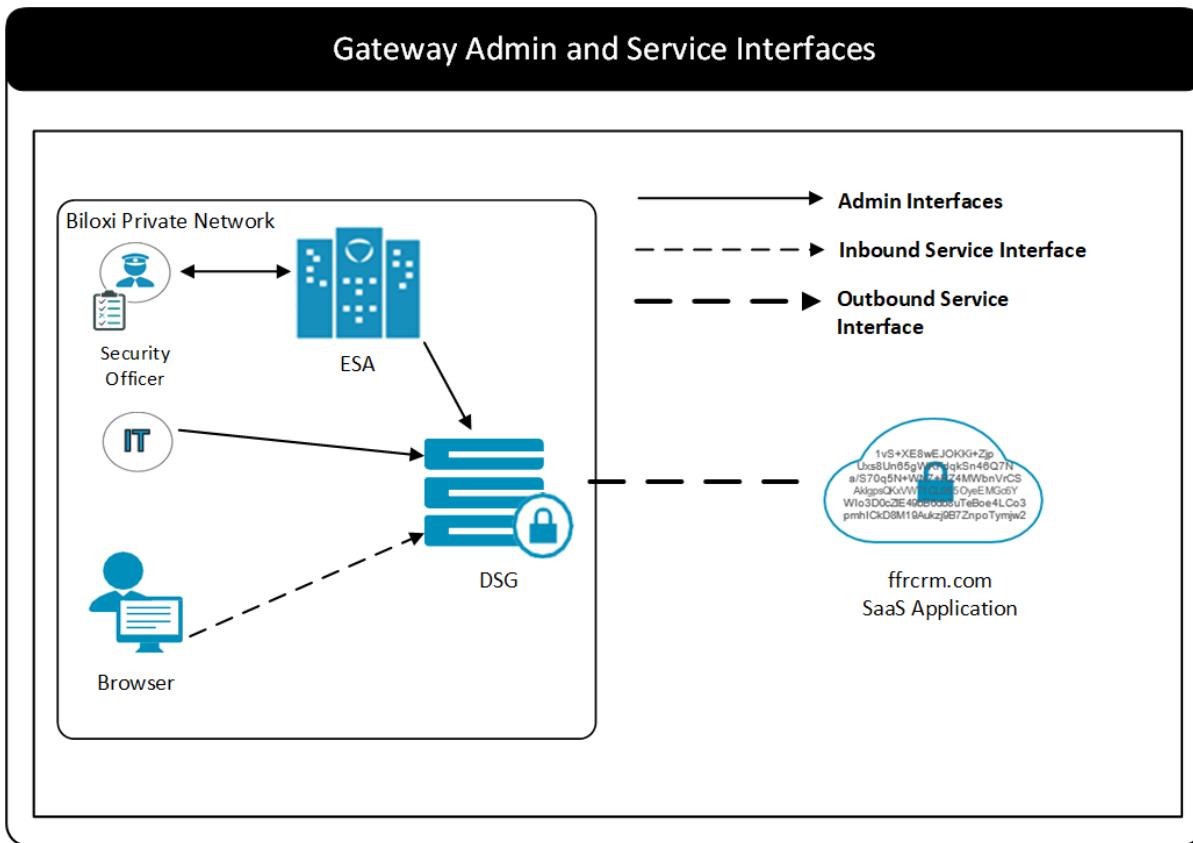


Figure 7-4: Gateway Admin and Services Interfaces

Network firewalls situated between consumer's gateway interfaces, admin or services, and between the gateway and the system it is expected to communicate with will require to adjust to allow it.

Note: The supported TLS versions are SSLv3, TLSv1.0, TLSv1.1, TLSv1.2, and TLSv1.3.

If you are utilizing the DSG appliance, the following ports must be configured in your environment.

Table 7-2: Ports for Users

Port Number/ TYPE (ECHO)	Protocol	Source	Destination	NIC	Description
22	TCP	System User	DSG	Management NIC (ethMNG)	Access to CLI Manager
443	TCP	System User	DSG	Management NIC (ethMNG)	Access to Web UI

The following are the list of ports that must be configured for communication between DSG and ESA.

Table 7-3: Ports for Communication with ESA

Port Number/ TYPE (ECHO)	Protocol	Source	Destination	NIC	Description	Notes (If any)
22	TCP	ESA	DSG	Management NIC (ethMNG)	<ul style="list-style-type: none"> • Replication or Rulesets from DSG to ESA • DSG Patching from ESA 	
443	TCP	ESA	DSG	Management NIC (ethMNG)	Communication in TAC	
443	TCP	DSG	ESA and Virtual IP address of ESA	Management NIC (ethMNG)	Downloading certificates from ESA	
8443	TCP	DSG	ESA and Virtual IP address of ESA	Management NIC (ethMNG)	<ul style="list-style-type: none"> • Establishing communication with ESA • Retrieving policy from ESA • Sending audit logs to ESA 	
389	TCP	DSG	Virtual IP address of ESA	Management NIC (ethMNG)	Authentication and authorization by ESA	
10100	UDP	DSG	ESA	Management NIC (ethMNG)	<ul style="list-style-type: none"> • Establishing communication with ESA • Communication in TAC 	This port is optional. If the appliance heartbeat services are stopped, this port can be disabled.
5671	TCP	DSG	Virtual IP address of ESA		Messaging between Protegility appliances.	While establishing communication with ESA, if the user notification is not set, you can disable this port.

The following are the list of ports that must also be configured when DSG is configured in a TAC.

Table 7-4: DSG Ports for Communication in TAC

Port Number/ TYPE (ECHO)	Protocol	Source	Destination	NIC	Description	Notes (If any)
22	TCP	DSG	ESA	Management NIC (ethMNG)	Communication in TAC	



Port Number/ TYPE (ECHO)	Protocol	Source	Destination	NIC	Description	Notes (If any)
8585	TCP	ESA	DSG	Management NIC (ethMNG)	Cloud Gateway cluster	
443	TCP	ESA	DSG	Management NIC (ethMNG)	Communication in TAC	
10100	UDP	ESA	DSG	Management NIC (ethMNG)	Communication in TAC	This port is optional. If the Appliance Heartbeat services are stopped, this port can be disabled.
10100	UDP	DSG	ESA	Management NIC (ethMNG)	<ul style="list-style-type: none"> Establishing communication with ESA Communication in TAC 	This port is optional. If the Appliance Heartbeat services are stopped, this port can be disabled.
10100	UDP	DSG	DSG	Management NIC (ethMNG)	Communication in TAC	This port is optional.
8300	TCP	ESA	DSG	Management NIC (ethMNG)	Used by servers to handle incoming request.	If your TAC utilizes Consul services, you must enable this port.
8300	TCP	DSG	DSG	Management NIC (ethMNG)	Handle incoming requests	If your TAC utilizes Consul services, you must enable this port.
8300	TCP	DSG	DSG	Management NIC (ethMNG)	Handle incoming requests	If your TAC utilizes Consul services, you must enable this port for communication between DSGs in a TAC.
8301	TCP and UDP	ESA	DSG	Management NIC (ethMNG)	Gossip on LAN.	If your TAC utilizes Consul services, you must enable this port.
8301	TCP and UDP	DSG	ESA	Management NIC (ethMNG)	Gossip on LAN.	If your TAC utilizes Consul services, you must open this port.
8301	TCP and UDP	DSG	DSG	Management NIC (ethMNG)	Gossip on LAN.	If your TAC utilizes Consul services, you must open this port.
8302	TCP and UDP	ESA	DSG	Management NIC (ethMNG)	Gossip on WAN.	If your TAC utilizes Consul services, you must enable this port.

Port Number/ TYPE (ECHO)	Protocol	Source	Destination	NIC	Description	Notes (If any)
8302	TCP and UDP	DSG	ESA	Management NIC (ethMNG)	Gossip on WAN.	If your TAC utilizes Consul services, you must enable this port.
8302	TCP and UDP	DSG	DSG	Management NIC (ethMNG)	Gossip on WAN.	If your TAC utilizes Consul services, you must enable this port.
8600	TCP and UDP	ESA	DSG	Management NIC (ethMNG)	Listens to the DNS server port.	If your TAC utilizes Consul services, you must enable this port.
8600	TCP and UDP	DSG	ESA	Management NIC (ethMNG)	Listens to the DNS server port.	If your TAC utilizes Consul services, you must enable this port.
9000	TCP and UDP	ESA	DSG	Management NIC (ethMNG)	Checks local certificates.	If your TAC utilizes Consul services, you must enable this port.
9000	TCP and UDP	DSG	ESA	Management NIC (ethMNG)	Checks local certificates.	If your TAC utilizes Consul services, you must enable this port.

Based on the firewall rules and network infrastructure of your organization, you must open ports for the services listed in the following table.

Table 7-5: Additional Ports for DSG

Port Number/ TYPE (ECHO)	Protocol	Source	Destination	NIC	Description	Notes (If any)
123	UDP	DSG	Time servers	Management NIC (ethMNG) of ESA	NTP Time Sync Port	You can change the port as per your organizational requirements.
514	TCP	DSG	Syslog servers	Management NIC (ethMNG) of ESA	Storing logs	You can change the port as per your organizational requirements.
N/A*	N/A*	DSG	Applications/ Systems	Service NIC (ethSRV) of DSG	Enabling communication for DSG with different applications in the organization	You can change the port as per your organizational requirements.
N/A*	N/A*	Applications/ System	DSG	Service NIC (ethSRV) of DSG	Enabling communication for DSG with different	You can change the port as per your

Port Number/ TYPE (ECHO)	Protocol	Source	Destination	NIC	Description	Notes (If any)
					applications in the organization	organizational requirements.

Note: N/A* - In DSG, service NICs are not assigned a specific port number. You can configure a port number as per your requirements.

7.7.1 NIC Bonding

The NIC is a device through which appliances, such as ESA or DSG, on a network connect to each other. If the NIC stops functioning or is under maintenance, the connection is interrupted, and the appliance is unreachable. To mitigate the issues caused by the failure of a single network card, Protegility leverages the NIC bonding feature for network redundancy and fault tolerance.

In NIC bonding, multiple NICs are configured on a single appliance. You then bind the NICs to increase network redundancy. NIC bonding ensures that if one NIC fails, the requests are routed to the other bonded NICs. Thus, failure of a NIC does not affect the operation of the appliance.

You can bond the configured NICs using different bonding modes.

Caution: The NIC bonding feature is applicable only for the DSG nodes that are configured on the on-premise platform. The DSG nodes that are configured on the cloud platforms, such as, AWS, Azure, or GCP, do not support this feature.

7.7.1.1 Bonding Modes

The bonding modes determine how traffic is routed across the NICs. The MII monitoring (MIIMON) is a link monitoring feature that is used for inspecting the failure of NICs added to the appliance. The frequency of monitoring is 100 milliseconds. The following modes are available to bind NICs together:

- Mode 0/Balance Round Robin
- Mode 1/Active-backup
- Mode 2/Exclusive OR
- Mode 3/Broadcast
- Mode 4/Dynamic Link Aggregation
- Mode 5/Adaptive Transmit Load Balancing
- Mode 6/Adaptive Load Balancing

The following two bonding modes are supported for appliances:

- **Mode 1/Active-backup policy:** In this mode, multiple NICs, which are slaves, are configured on an appliance. However, only one slave is active at a time. The slave that accepts the requests is active and the other slaves are set as standby. When the active NIC stops functioning, the next available slave is set as active.
- **Mode 6/Adaptive load balancing:** In this mode, multiple NICs are configured on an appliance. All the NICs are active simultaneously. The traffic is distributed sequentially across all the NICs in a round-robin method. If a NIC is added or removed from the appliance, the traffic is redistributed accordingly among the available NICs. The incoming and outgoing traffic is load balanced and the MAC address of the actual NIC receives the request. The throughput achieved in this mode is high as compared to mode 1.

7.7.1.2 Prerequisites

Ensure that you complete the following pre-requisites when binding interfaces:



- The IP address is assigned only to the NIC on which the bond is initiated. You must not assign an IP address to the other NICs.
- The NICs are on the same network.

7.7.1.3 Creating a Bond

This section describes the procedure to create a bond between NICs.

Note: Ensure that the IP address of the slave nodes are static.

Note: Ensure that you have added a default gateway for the Management NIC (ethMNG) and Service NIC (ethSRV0). For more information about adding a default gateway to the Management NIC and Service NIC, refer to the section [Configuring Default Gateway for Network Interfaces](#).

Note: When a bond is created with any service NIC (ethSRVX) in the Web UI, its status indicator appears red - which may indicate it is not functioning properly - even though the service NIC (ethSRVX) is active. To change the service NIC (ethSRVX) status indicator to green, click **Refresh**.

► To create a bond:

1. On the DSG Web UI, navigate to **Settings > Network > Network Settings**.
The *Network Settings* screen appears.
2. Under the Network Interfaces area, click **Create Bond** corresponding to the interface on which you want to initiate the bond.
The following screen appears.

Create Network Teaming : ethMNG (00:50:56:01:1f:39 (VMware, Inc.))

Network bonding policy mode :	Active-backup policy	
<input type="checkbox"/> Name	MAC ID	
<input type="checkbox"/> ethSRV8	00:50:56:01:26:84 (VMware, Inc.)	
<input type="checkbox"/> ethSRV7	00:50:56:01:26:83 (VMware, Inc.)	
<input type="checkbox"/> ethSRV6	00:50:56:01:26:1b (VMware, Inc.)	
<input type="checkbox"/> ethSRV5	00:50:56:01:26:1a (VMware, Inc.)	
<input type="checkbox"/> ethSRV4	00:50:56:01:23:1a (VMware, Inc.)	
<input type="checkbox"/> ethSRV3	00:50:56:01:22:d3 (VMware, Inc.)	
<input type="checkbox"/> ethSRV2	00:50:56:01:22:d2 (VMware, Inc.)	

Figure 7-5: Creating a Bond

Note: Ensure that the IP address is assigned to the interface on which you want to initiate the bond.

3. Select the following modes from the drop down list:
 - *Active-backup policy*
 - *Adaptive Load Balancing*
4. Select the interfaces with which you want to create a bond.
5. Select **Establish Network Bonding**.
A confirmation message appears.
6. Click **OK**.

The bond is created and the list appears on the Web UI.

7.7.1.4 Removing a Bond

The following procedure describes the steps to remove a bond between NICs.

► To remove a bond:

1. On the DSG Web UI, navigate to **Settings > Network > Network Settings**.

The *Network Settings* screen appears with all the created bonds as shown in the following figure.

IP/Netmask	Link	DHCP	Network Teaming 	Addresses
ethMNG 00:50:56:01:1f:da (VMware, Inc.) 		<input type="checkbox"/>	Slaves : ethSRV0 Mode : 1 Remove Bond	2.10.1.2/255.255.252.0 Edit
ethSRV1 00:50:56:01:1f:de (VMware, Inc.) 		<input checked="" type="checkbox"/>	Not Available 	10.10.96.189/255.255.240.0
ethSRV2 00:50:56:01:1f:ea (VMware, Inc.) 		<input checked="" type="checkbox"/>	Slaves : ethSRV3 Mode : 6 Remove Bond	10.10.96.193/255.255.240.0

- Under the *Network Interfaces* area, click **Remove Bond** corresponding to the interface on which the bonding is created. A confirmation screen appears.

- Select **OK**.

The bond is removed and the interfaces are visible on the *IP/Network* list.

7.7.1.5 Viewing a Bond

Using the DSG CLI Manager, you can view the bonds that are created between all the interfaces.

► To view a bond:

- On the DSG CLI Manager, navigate to **Networking > Network Settings**.

The *Network Configuration Information Settings* screen appears.

- Navigate to **Interface Bonding** and select **Edit**.

The *Network Teaming* screen displaying all the bonded interfaces appears as shown in the following figure.

IP/Netmask	Link	DHCP	Network Teaming 	Addresses
ethMNG 00:50:56:01:1f:da (VMware, Inc.) 		<input type="checkbox"/>	Slaves : ethSRV0 Mode : 1 Remove Bond	2.10.1.2/255.255.252.0 Edit
ethSRV1 00:50:56:01:1f:de (VMware, Inc.) 		<input checked="" type="checkbox"/>	Not Available 	10.10.96.189/255.255.240.0
ethSRV2 00:50:56:01:1f:ea (VMware, Inc.) 		<input checked="" type="checkbox"/>	Slaves : ethSRV3 Mode : 6 Remove Bond	10.10.96.193/255.255.240.0

7.7.1.6 Resetting the Bond

You can reset all the bonds that are created for an appliance. When you reset the bonds, all the bonds created are disabled. The slave NICs are reset to their initial state, where you can configure the network settings for them separately.

► To reset all the bonds:

1. On the DSG CLI Manager, navigate to **Networking > Network Settings**.

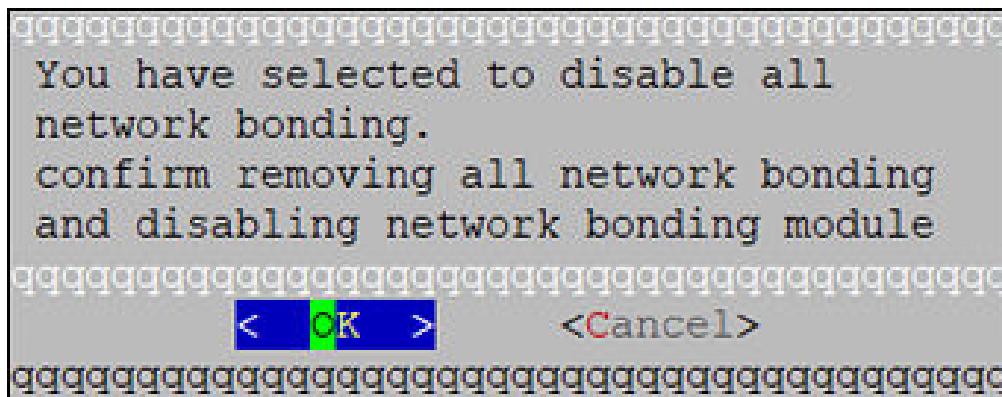
The *Network Configuration Information Settings* screen appears.

2. Navigate to **Interface Bonding** and select **Edit**.

The *Network Teaming* screen displaying all the bonded interfaces appears.

3. Select **Reset**.

The following screen appears.



4. Select **OK**.

The bonding for all the interfaces is removed.

7.8 HTTP URL Rewriting

Operating in the in-band mode of data protection against SaaS applications, DSG is placed between the end-user's client devices and the SaaS servers on the public Internet. For DSG to intercept the traffic between end-user devices and SaaS servers, the top level public Internet Fully Qualified Domain Names (FQDN) that are made accessible by the SaaS need to be identified. Once identified, these FQDNs shall be mapped to internal URLs pointed at DSG and the corresponding URL mappings shall be configured in DSG.

Like most websites on the public Internet, SaaS applications are accessed by their users through one or more Uniform Resource Locators (URL). These are typically HTTP(S) URLs that are made up of FQDNs, e.g. <https://www.ffcrm.com>, which are uniquely routable on the public Internet. A SaaS may be accessible on the public Internet through many public facing URLs. An identification of all such public URLs is essential for ensuring that all the traffic between the end users and the SaaS can be routed through DSG. A list of top level Internet facing FQDNs of a SaaS may be gathered from the following sources:

- **SaaS Support Documentation:** SaaS providers typically provide publicly available documentation where they publish their externally visible FQDNs. This information typically exists for the IT teams of customer enterprises such that they can allow (white list) access to these FQDNs through their corporate firewalls.
- **Using Browser Tools or Network Sniffers:** As an alternative or in addition, the IT team at Biloxi Corp may attempt to find the public FQDNs of [ffcrm.com](https://www.ffcrm.com) themselves. This can be achieved by making use of network sniffers (possibly an embedded function within Biloxi Corp's corporate firewall or a forward proxy).

An alternative is to use ‘developer tools’ in the user’s web browser. Browser developer tools show a complete trace of HTTP messaging between the browser and the SaaS. If all relevant sections of ffcrm.com SaaS have been accessed, this trace will reveal the relevant public FQDNs made visible by ffcrm.com.

As a result of performing the above steps, let’s consider that the IT team at Biloxi Corp has identified the following top level public FQDNs exposed by ffcrm.com.

- www.ffcrm.com
- login.ffcrm.com
- crm.ffcrm.com
- analytics.ffcrm.com

For DSG to interwork traffic between its two sides (end users and SaaS), DSG relies on FQDN translation. The following figure shows FQDN translation performed by DSG.

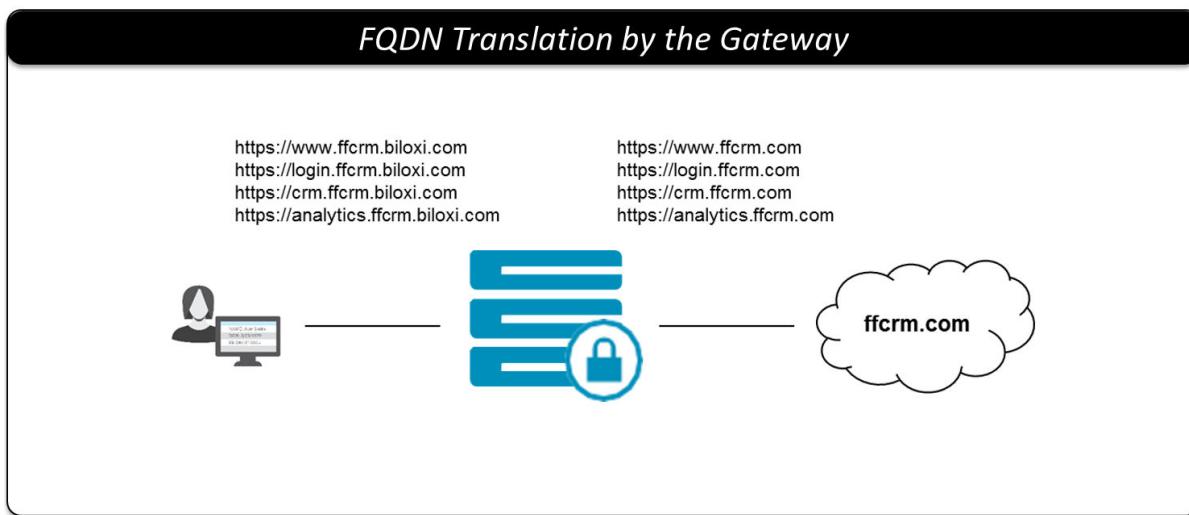


Figure 7-6: FQDN Translation by the Gateway

The above domain names will be mapped to internal domain names pointed at DSG. For instance, DSG will be configured with following URL mappings.

Table 7-6: Domain Name Mapping

Incoming Request URL	Outgoing Request URL
https://www.ffcrm.biloxi.com	https://www.ffcrm.com
https://login.ffcrm.biloxi.com	https://login.ffcrm.com
https://crm.ffcrm.biloxi.com	https://crm.ffcrm.com
https://analytics.ffcrm.biloxi.com	https://analytics.ffcrm.com

This domain name mapping can be generalized by configuring a Domain Name Service (DNS) with a global mapping of *.ffcrm.biloxi.com to DSG which will apply to any of the sub domain www, login, crm, analytics or any other that might be added by ffcrm.com in the future.

Ultimately, end users will be consuming the service through the internal host names. Techniques like Single Sign On (SSO) using Security Assertion Markup Language (SAML) can be used to force users to use internal host names even if direct access to the external ones is attempted.

7.9 Clustering and Load Balancing

DSG deployed as a cluster of appliance nodes provides the necessary overall system capacity as well as high availability through redundancy.

Note: Nodes within a DSG cluster operate autonomously in an active/active arrangement. This is in contrast with an active/passive arrangement seen in ESA High-Availability where the participating nodes communicate with each other to synchronize state such as configuration.

Dependent on capabilities of underlying server hardware, traffic patterns and a few other factors, a single DSG node can process a certain amount of traffic. The size of a DSG cluster is determined by comparing the capacity of single node against customer's performance requirements.

Note: For more information about the specific metrics collected in a controlled performance test environment, contact Protegility Support for DSG Performance Report.

Let's consider that the IT team at Biloxi Corp has established that they need three DSG nodes to meet their capacity and availability requirements. To hide DSG cluster topology from the end-users, the cluster is fronted by an off-the-shelf Load Balancer.

While considering load-balancing of HTTP traffic, since DSG nodes are stateless in and of themselves and across HTTP transactions, DSG places minimum requirements on Load Balancers (LBs). For instance, LBs fronting DSG cluster are not required maintain session stickiness or affinity. In fact, these LBs may be configured to operate at the lowest layers of TCP/IP protocol stack such as the networking or transport while being unaware of the application layer (HTTP).

Note: When available, DSG logging will leverage X-Real-IP HTTP Header added by Load Balancers to represent the actual client address.

The following figure shows a DSG cluster comprised of three gateway nodes fronted by a Load Balancer.

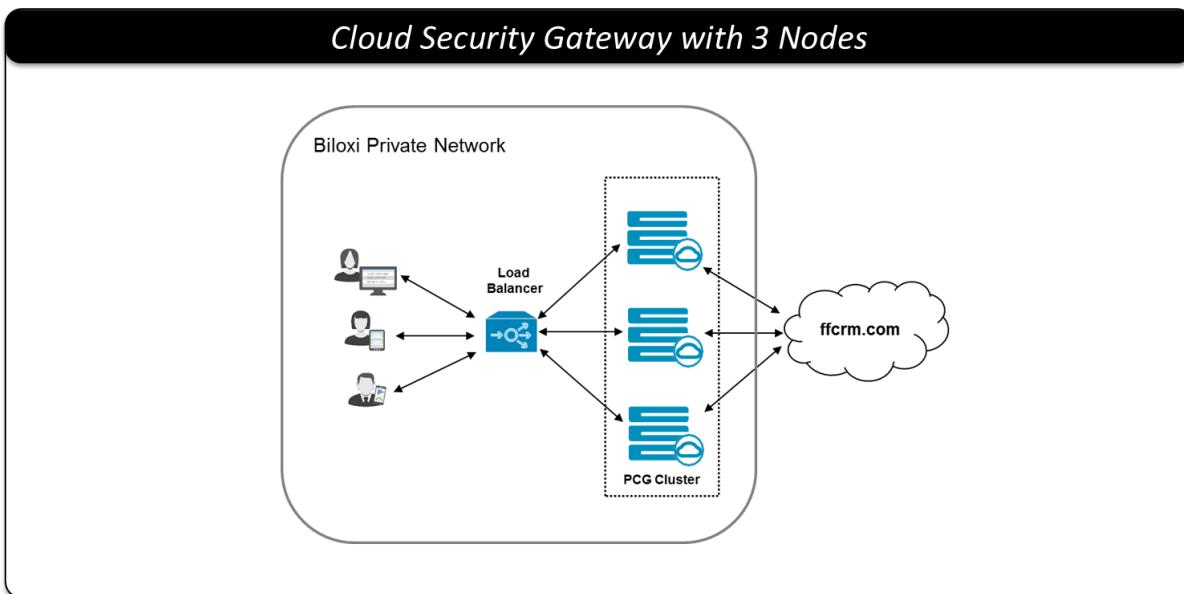


Figure 7-7: Cloud Security Gateway with 3 Nodes

7.10 SSL Certificates

The use of secured socket layer (aka SSL) prevents a man-in-the-middle from tampering or eavesdropping the communication between two parties. Though it may not be a requirement it is certainly a best practice to secure all communication channels that may be used to transmit sensitive data. DSG function is to transform data transmitted through it. To achieve that over a secured communication channel it is necessary for DSG to terminate the inbound TLS/SSL communication. This step may be skipped when no inbound SSL is used, otherwise, SSL Server Certificate and Keys are needed for DSG to properly terminate inbound SSL connections.

During the install process of DSG, a series of self-signed SSL Certificates are generated for your convenience. You may use it in non-production environment. It is recommended however to use your own certificate for production use.

No further action is required if you choose to use the service certificate generated during install time.

Certificate and keys can be uploaded for use with DSG cluster after the installation. Should you choose to use certificates generated elsewhere, be prepared to upload both the certificate and the associated key in such case. Supported certificate file formats are *.crt* and *.pem*.

You may need to generate your own self-signed certificate of specific attributes such as hostname, key strength or expiration date.

Chapter 8

Installing the DSG

- [8.1 Installing the DSG On-Premise](#)
- [8.2 Installing the DSG on Cloud Platforms](#)
- [8.3 Extending ESA with DSG Web UI](#)
- [8.4 Setting up ESA Communication](#)
- [8.5 Configuring Default Gateway for Network Interfaces](#)
- [8.6 Configuring the DSG Cluster](#)
- [8.7 Forwarding Logs to the Audit Store](#)
- [8.8 Advanced Settings](#)

The DSG can be installed as an on-premise appliance or on cloud platforms. This section provides information about installing the DSG on these platforms.

8.1 Installing the DSG On-Premise

The Data Security Gateway (DSG) install requires an existing ESA, which serves as a single point of management for the data security policy, rules configuration, and on-going monitoring of the system. This section provides information about the recommended order of the steps to install a DSG appliance.

Before you begin:

- Ensure that an ESA 9.1.0.5 is installed.

Note:

For more information about installing the ESA 9.1.0.5, refer to the sections *Installing Appliance On-Premise* and *Installing Appliances on Cloud Platforms* in the [Protegility Installation Guide 9.1.0.5](#).

- Ensure that the *ESA_PAP-ALL-64_x86-64_9.1.0.5.xxxx.DSGUP.pty* patch is available. This patch is applied on the ESA to extend the ESA with the DSG Web UI.
- Ensure that the *DSG_PAP-ALL-64_x86-64_3.1.0.5.x.iso* image is available. The *.iso* image is used to install the DSG appliance.
- Ensure that the *DSG_PAP-ALL-64_x86-64_3.1.0.5.x.UPtgz* file is available. This file contains the *DSG_PAP-ALL-64_x86-64_3.1.0.5.x.UP.pty* and *PAP_PAP-ALL-64_x86-64_9.1.0.5.x.UP.pty* patch files.

Ensure that you use the following installation order to install the DSG on the ESA.

Table 8-1: Order of Installation on the ESA

Order of installation	Description	Affected Appliance	Reference
1	<p>Apply the DSG v3.1.0.5 patch (<i>ESA_PAP-ALL-64_x86-64_9.1.0.5.xxxx-DSGUP.pty</i>) on the ESA v9.1.0.5.</p> <p>Note: Before applying a patch on the ESA, it is recommended to take a full OS backup from the ESA Web UI. For more information about taking a full OS backup from the ESA Web UI, refer to the section <i>Backing up Appliance OS in the Web UI</i> in the <i>Protegility Appliances Overview Guide 9.1.0.5</i>.</p>	ESA	Extending ESA with DSG Web UI
2	Create a Trusted Appliance Cluster (TAC) on the ESA.	ESA	Create a TAC
3	Initialize PIM	ESA	Section <i>Initializing the Policy Management</i> in the <i>Protegility Policy Management Guide 9.1.0.5</i> .
4	Initialize Analytics	ESA	Section <i>Initializing the Audit Store Cluster on the ESA</i> in the <i>Protegility Installation Guide 9.1.0.5</i> .

Note: By default, the *default_80* HTTP tunnel is disabled for the security reasons. If you want to use *default_80* HTTP tunnel in any service, then you must enable the *default_80* HTTP tunnel from the Web UI.

To enable the *default_80* HTTP tunnel, on the ESA Web UI, navigate to **Cloud Gateway > Transport**, and click the **Tunnels** tab. Select the **default_80** HTTP tunnel and click **Edit**.

After the *default_80* tunnel is enabled, you must restart the gateway. On the **Tunnels** tab, click **Deploy to All Nodes** to restart the gateway.

Ensure that you use the following installation order to install the DSG.

Table 8-2: Order of Installation on the DSG

Order of installation	Description	Affected Appliance	Reference
1	Install the DSG v3.1.0.5 ISO.	DSG	Installing DSG
2	Configure the Default Gateway for the Service NIC using the DSG CLI Manager.	DSG	Configuring Default Gateway for Service NIC (ethSRV0) using the DSG CLI Manager
3	Add the DSG nodes to the existing Trusted Appliance Cluster from the Cluster tab.	ESA	Adding a DSG node
4	Configure the DSG to forward the logs to the <i>Audit Store</i> on the ESA.	DSG	Forwarding Logs to the Audit Store

8.1.1 Installing DSG

This section provides information about installing the DSG ISO.



Before you begin

- Ensure that **HubController** service is running on the ESA.

Caution:

Ensure that the Policy Management (PIM) has been initialized on the ESA. The initialization of PIM ensures that cryptographic keys for protecting data and the policy repository have been created.

For more information about initializing the PIM, refer to the section *Initializing the Policy Management* in the *Protegility Policy Management Guide 9.1.0.5*.

Caution: Ensure that the **Analytics** is initialized on the ESA. The initialization of the **Analytics** displays the information from the Audit Store.

For more information about initializing the **Analytics**, refer to the section *Initializing the Audit Store Cluster on the ESA* in the *Protegility Installation Guide 9.1.0.5*.

► To install the DSG:

1. Insert and mount the DSG installation media (ISO).
2. Restart the machine and ensure that it boots up from the installation media.

The following *DSG* splash screen appears.



3. Press **ENTER** to start the install procedure.
4. The following screen appears only when you reimagine to the DSG v3.1.0.5 from any older DSG version. You must enter **YES** to proceed with the installation, press **Tab** to select **OK**, and then press **ENTER**.



Figure 8-1: Reimage installation screen

- The following installation options screen appears. Select **INSTALL Destroy content and install a new copy**, press **Tab** to select **OK**, and then press **ENTER**.

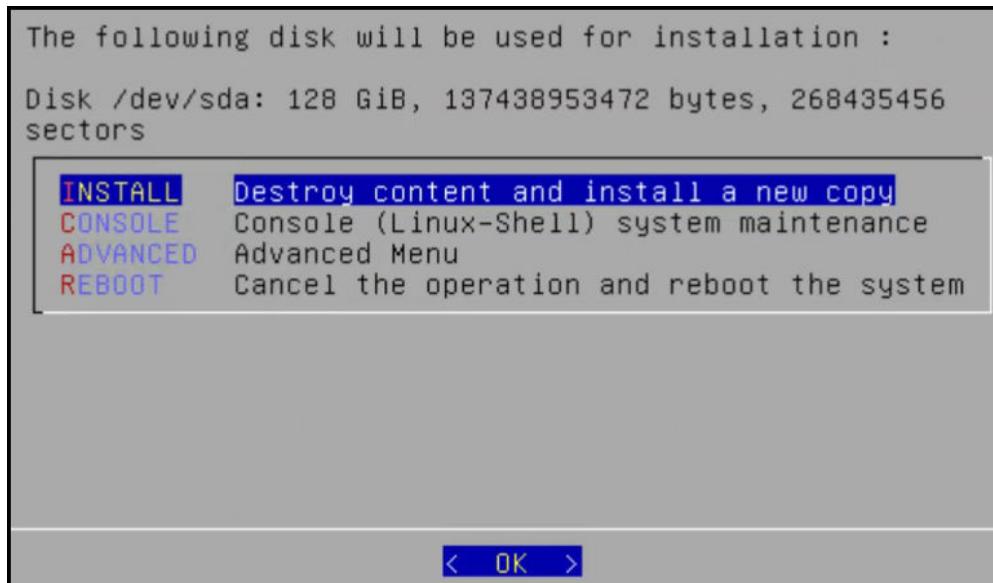


Figure 8-2: Installation Options screen

Note: If an existing file system is detected on the host hard drive, then a warning may appear prior to this step. The users who choose to proceed with the install will be prompted with additional confirmation before the data on the file system is lost.

- The system restarts and the *Network interfaces* screen with the network interfaces detected appears.
- Press **Tab** to select the network interface to be used for managing, which will be the management NIC for the DSG node, and proceed by pressing **Tab** to select **Select** and press **Enter**.

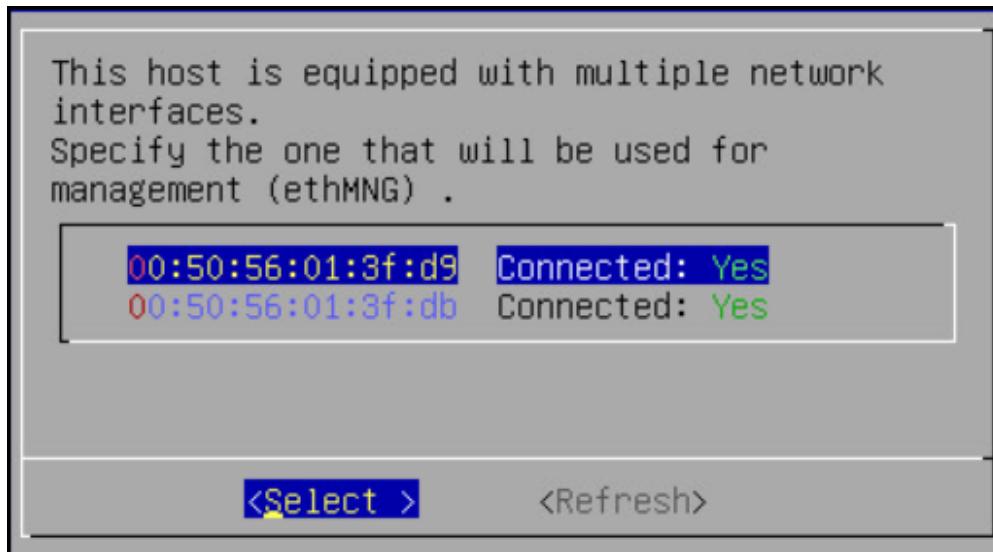


Figure 8-3: Network interface screen

Note: The selected network interface will be used for communication between the ESA and the DSG.

- The DSG appliance attempts to detect a DHCP server to setup the network configuration. If the DHCP server is detected, then the *Network Configuration Information* screen appears with the settings provided by the DHCP server. If you want to modify the auto-detected settings, then press **Tab** to select **Edit** and press **Enter** to update the information.

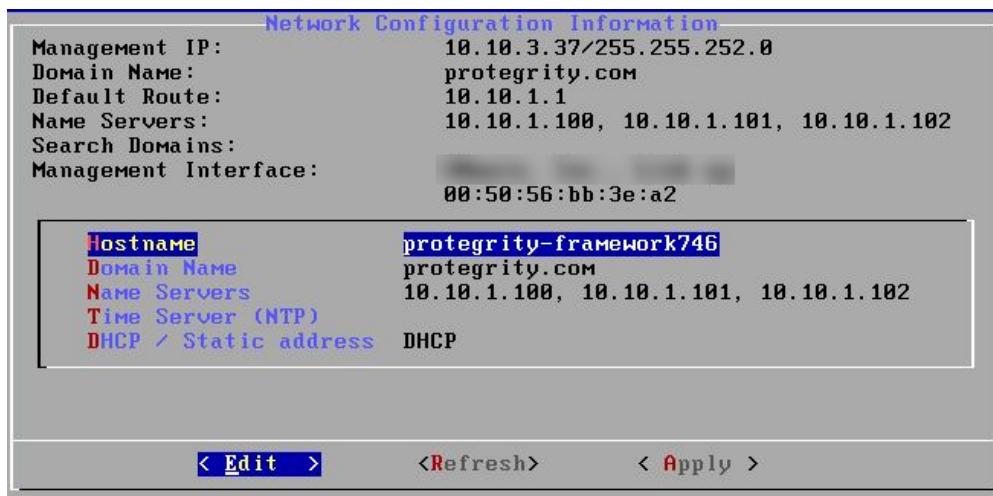


Figure 8-4: Network Configuration Information Screen

- Press **Tab** to select **Apply** and press **ENTER**.

Note: If a DHCP server is detected, then the *Select a node* screen appears. Select the ESA IP address that you want to use for the ESA communication from the list.

The following dialog appears when the DHCP server is not detected. Press **Tab** to select **Manual**, and press **Enter** to provide the IP address manually or **Retry** to attempt locating the DHCP server again.

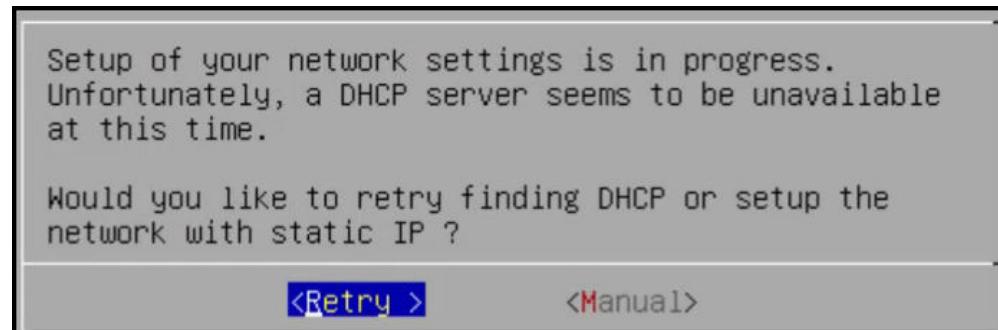


Figure 8-5: Setting up network manually

The **Network Configuration Information** dialog appears. You must enter the network configuration and select **Apply** to continue.



Figure 8-6: Network Configuration Information

Note: On the DSG, the Management Interface is used for communication between the ESA and the DSG, and accessing the DSG Web UI. The Service Interface is used for handling the network traffic traversing through the DSG.

For more information about the management interface and the service interface, refer to the section [Network Planning](#).

9. Press **Tab** to select the time zone of the host, press **Tab** to select **Next**, and then press **ENTER**.



Figure 8-7: Time Zone screen

10. Press **Tab** to select the nearest location, press **Tab** to select **Next**, and then press **ENTER**.

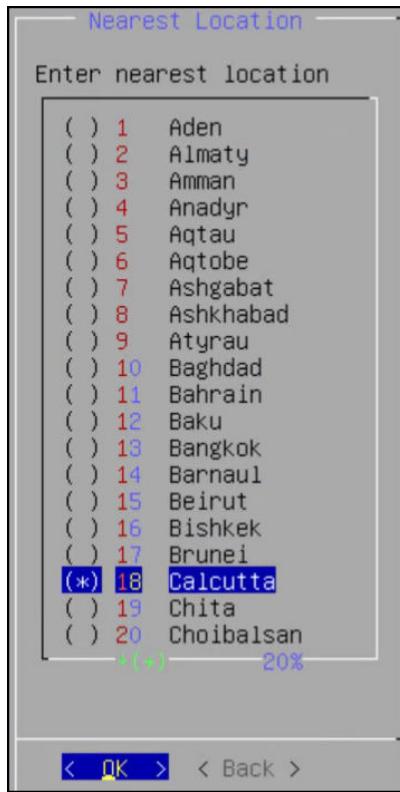


Figure 8-8: Nearest Location screen

11. Press **Tab** to select required option, press **Tab** to select **OK**, and then press **ENTER**.

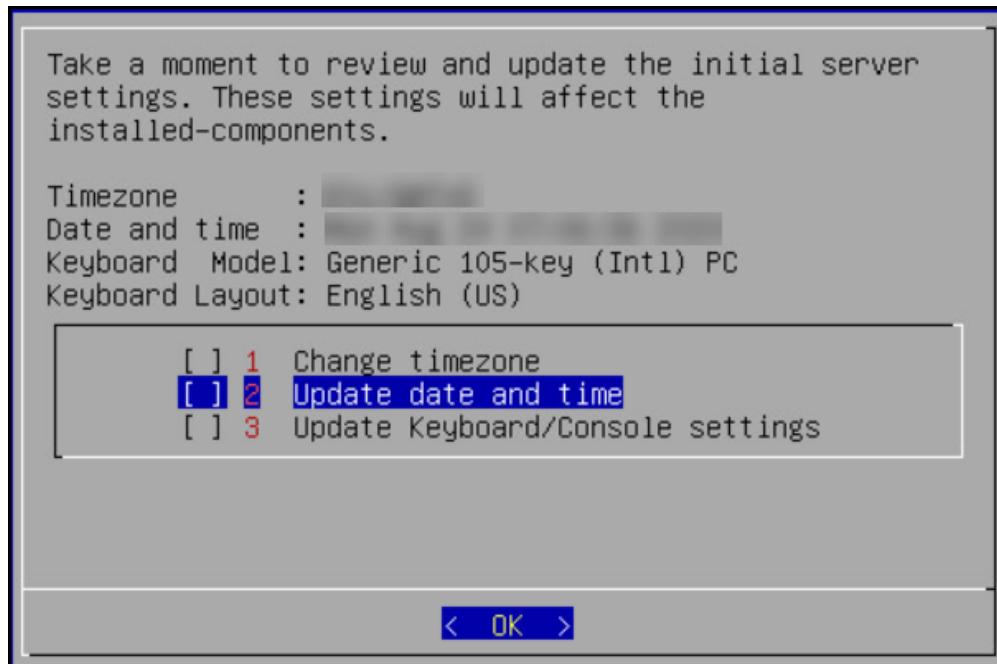


Figure 8-9: Update Date/Time Zone screen

12. Press **Tab** to select required option and then press **ENTER**.

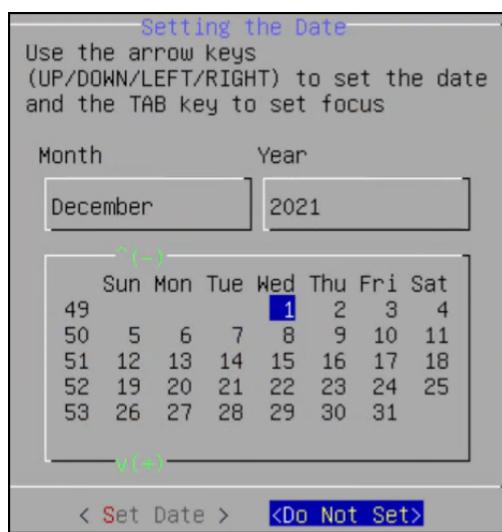


Figure 8-10: Setting the date screen

13. Press **Tab** to select required option and then press **ENTER**.

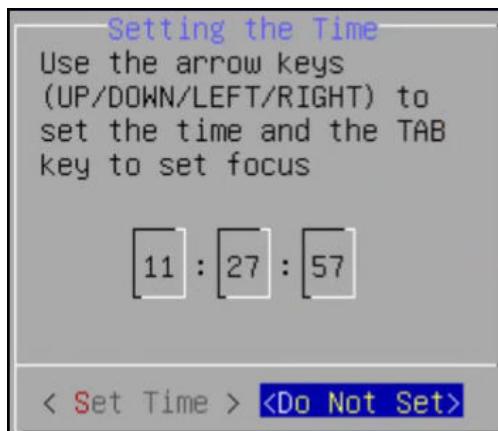


Figure 8-11: Setting the time screen

14. Select **Enable**, press **Tab** to select **OK**, and then press **ENTER** to provide the credentials for securing the GRand Unified Bootloader (GRUB).

Note:

GRUB is used to provide enhanced security for the DSG appliance using a username and password combination.

For more information about using GRUB, refer to the section *Securing the GRand Unified Bootloader (GRUB)* in the [Protegility Appliances Overview Guide 9.1.0.5](#).

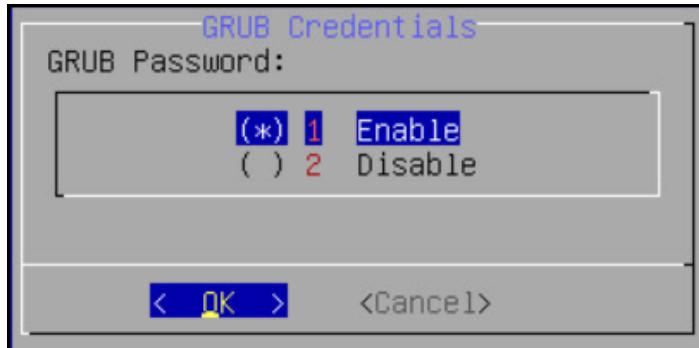


Figure 8-12: Enabling GRUB credentials

Caution: The *GRUB* option is available only for on-premise installations.

15. Enter a username, password, and password confirmation on the screen, select **OK** and press **ENTER**.

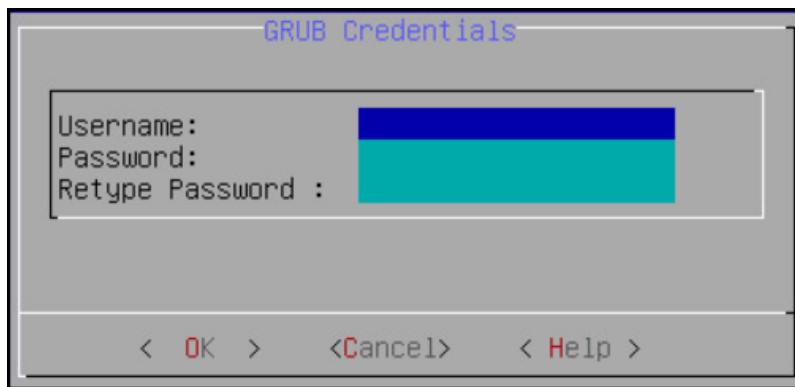


Figure 8-13: GRUB Credentials screen

Note:

The requirements for the **Username** are as follows:

- It should contain a minimum of three and maximum of 16 characters
- It should not contain numbers and special characters

Note:

The requirements for the **Password** are as follows:

- It must contain at least eight characters
- It must contain a combination of alphabets, numbers, and printable characters

16. Press **Tab** to set the user passwords, and then press **Tab** to select **Apply** and press **Enter**.



Figure 8-14: User Passwords screen

Note: It is recommended that strong passwords are set for all the users.

For more information about password policies, refer to the section *Strengthening Password Policy* in the *Protegility Appliances Overview Guide 9.1.0.5*.

17. Enter the IP address or hostname for the ESA. Press **Tab** to select **OK** and press **ENTER**. You can specify multiple IP addresses separated by comma.

The *Forward Logs to Audit Store* screen appears.



Figure 8-15: Forward Logs to Audit Screen

Note:

If the IP address or hostname of ESA is not provided while installing the DSG, then the user can add the ESA through [ESA Communication](#).

18. Select the ESA that you want to connect with, and then press **Tab** to select **OK** and press **ENTER**.

The *ESA Selection* screen appears.

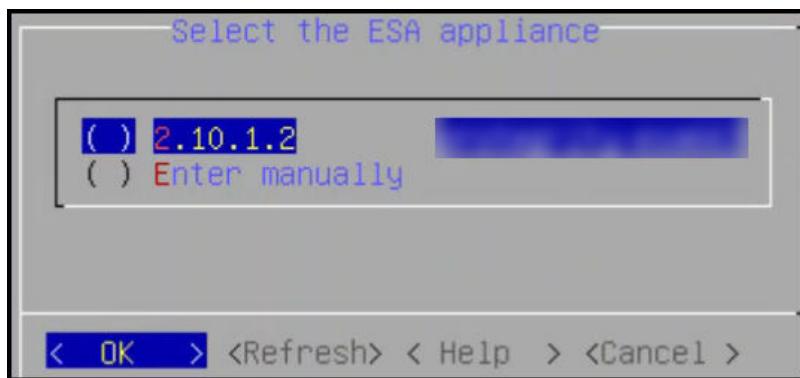


Figure 8-16: ESA appliance selection screen

Note: If you want to enter the ESA details manually, then select the **Enter manually** option. You must enter the ESA IP address when this option is selected.

19. Provide the username and password for the ESA that you want to communicate with, press **Tab** to select **OK**, and then press **ENTER**.

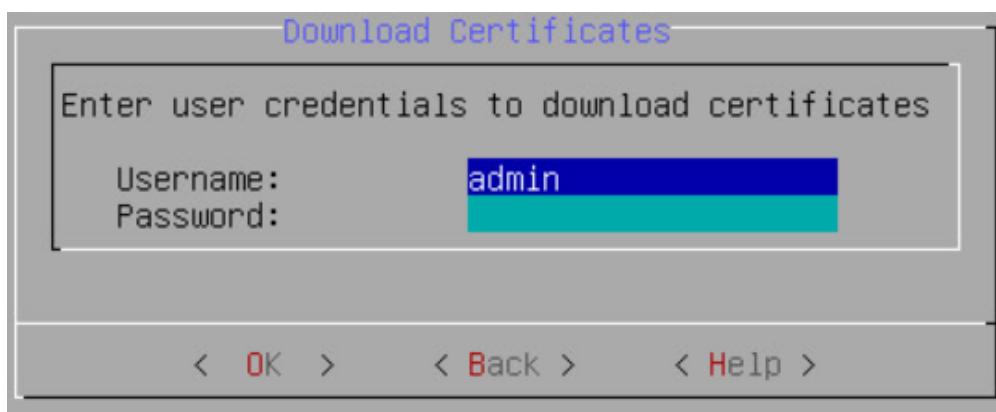


Figure 8-17: Download Certificates dialog

20. Enter the **IP Address** and **Network Mask** to configure the service interface and press **Tab** to select **OK** and press **ENTER**.

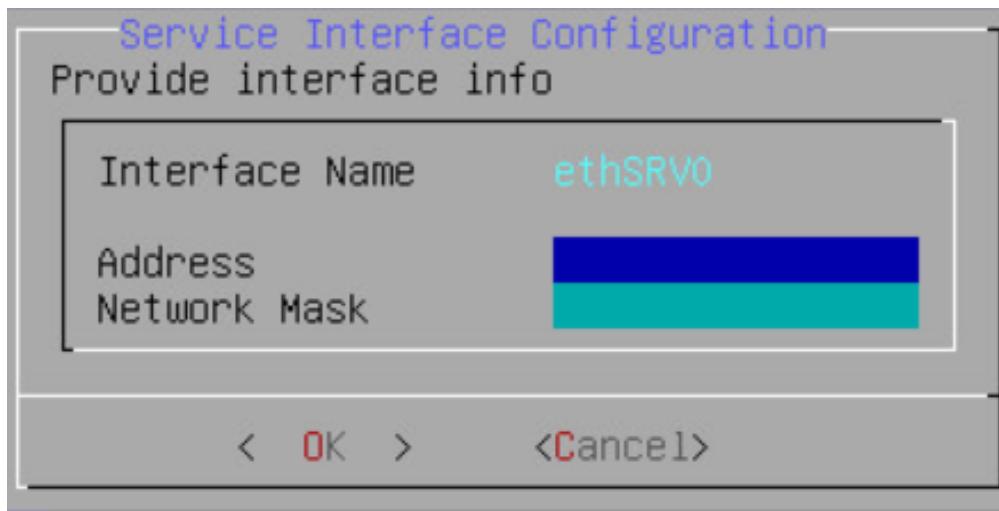


Figure 8-18: Service Interface Configuration screen

Caution: For ensuring network security, the DSG isolates the management interface from the service interface by allocating each with a separate network address. Ensure that two NICs are added to the DSG.

21. Select the **Cloud Utility AWS v2.1.3**, press **Tab** to select **OK**, and then press **ENTER** to install the utility. The **Cloud Utility AWS v2.1.3** utility must be selected if you plan to forward the DSG logs to AWS CloudWatch. If you choose to install the **Cloud Utility AWS v2.1.3** utility later, you can install this utility from the DSG CLI using the **Add or Remove Services** option after installing the DSG.

Note:

For more information about forwarding the DSG logs to AWS CloudWatch, refer to the section [Forwarding logs to AWS CloudWatch](#).

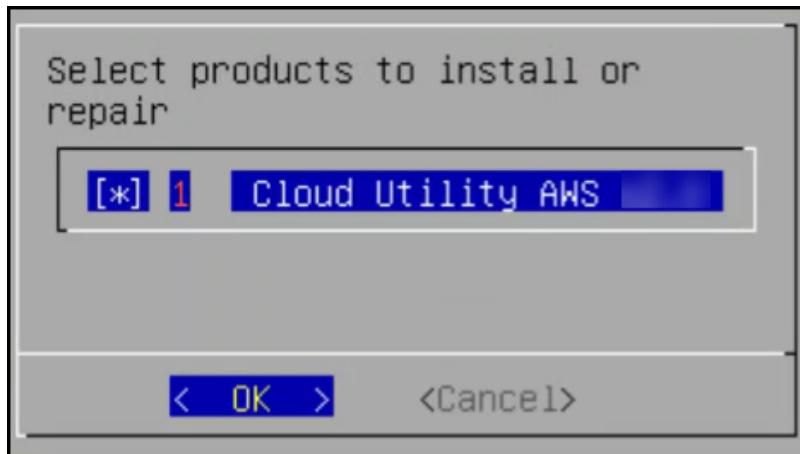


Figure 8-19: Select products to install screen

22. Select all the options except **Join Cloud Gateway Cluster**, press **Tab** to select **Set Location Now** and press **ENTER**.

Caution:

Ensure that the **Join Cloud Gateway Cluster** option is deselected. It is recommended that the user adds the DSG node from the **Cluster** tab after the DSG installation is completed.

For more information about adding a node to cluster, refer to the section [Adding a DSG node](#).

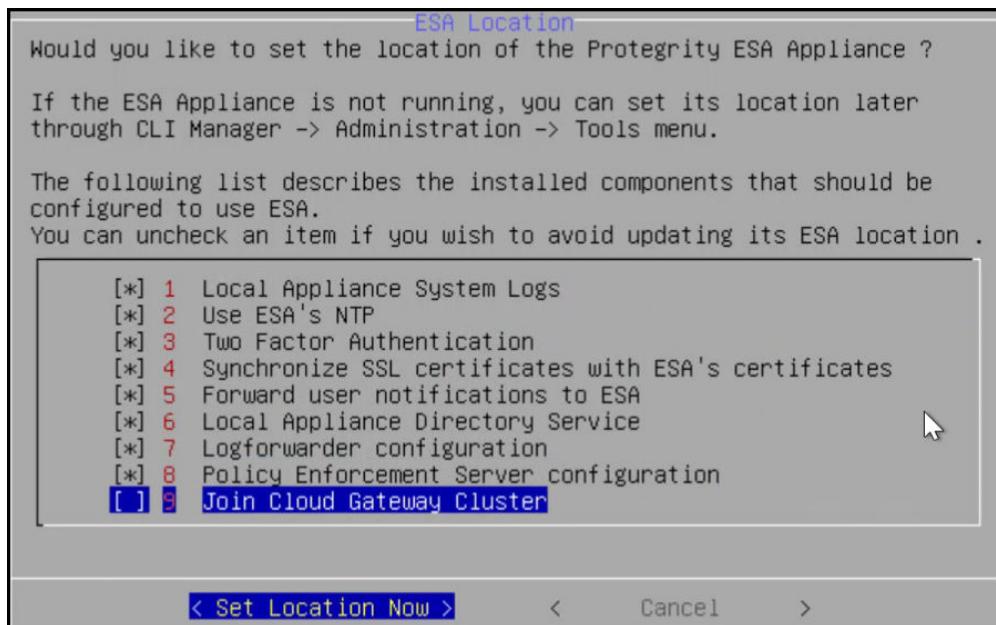


Figure 8-20: ESA Location screen

23. Select the ESA that you want to connect with, and then press **Tab** to select **OK** and press **ENTER**.
The *ESA selection* screen appears.

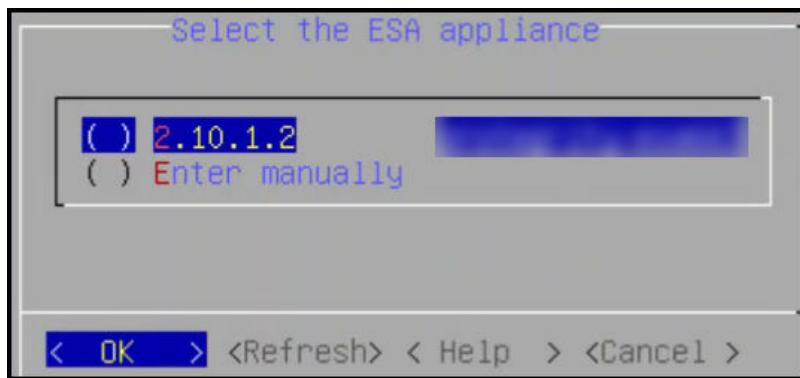


Figure 8-21: ESA appliance selection screen

Note: If you want to enter the ESA details manually, then select the **Enter manually** option. You will be asked to enter the ESA IP address or hostname when this option is selected.

24. Enter the ESA administrator username and password to establish communication between the ESA and the DSG. Press **Tab** to select **OK** and press **Enter**.

The *Enterprise Security Administrator - Admin Credentials* screen appears.

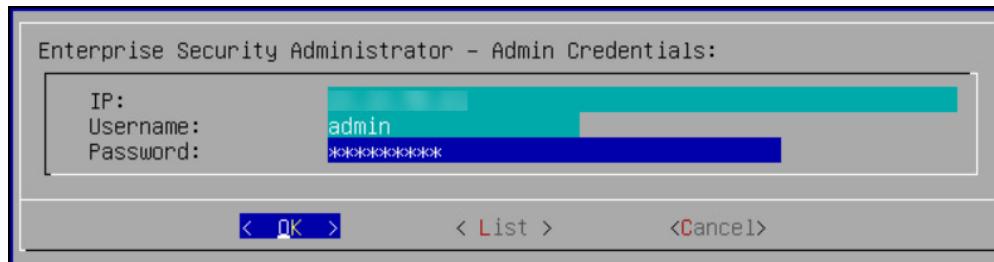


Figure 8-22: Enterprise Security Administrator - Admin Credentials screen

25. Enter the IP address or hostname for the ESA. Press **Tab** to select **OK** and press **ENTER**. You can specify multiple IP addresses separated by comma.

The *Forward Logs to Audit Store* screen appears.

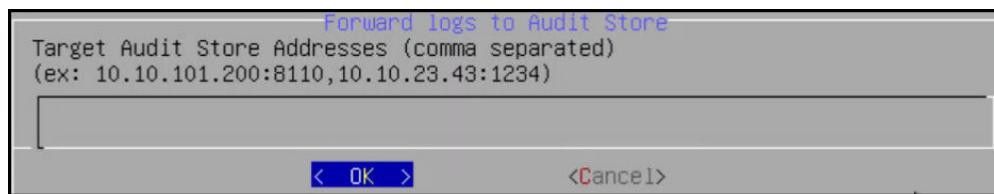


Figure 8-23: Forward Logs to Audit Screen

26. After successfully establishing the connection with the ESA, the following Summary dialog box appears.

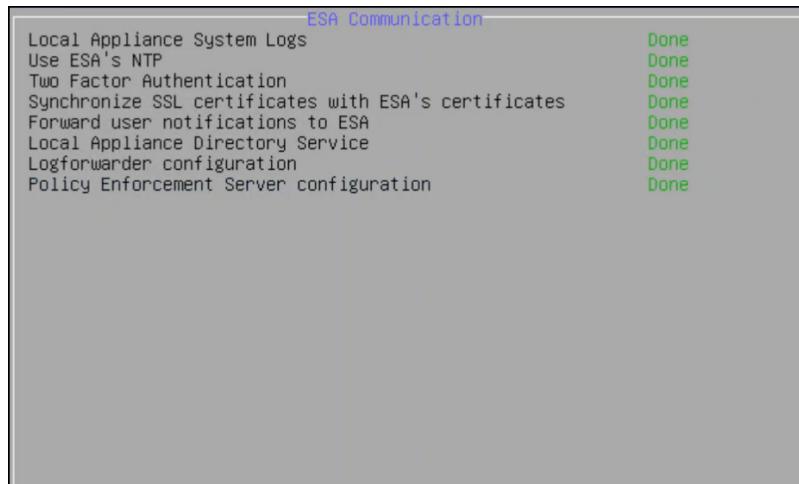


Figure 8-24: ESA Communication - Summary screen

27. Press **Tab** to select **Continue** and press **Enter** to continue to the DSG CLI manager.

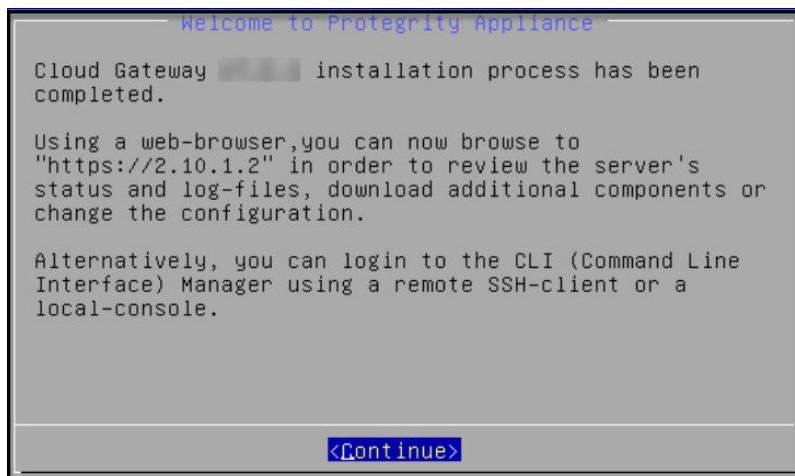


Figure 8-25: Welcome Screen

A Welcome to Protegility Appliance dialog box appears.

28. Login to the DSG CLI Manager.
29. Navigate to **Administration > Reboot and Shutdown**.
30. Select **Reboot** and press **Enter**.
31. Provide a reason for restarting the DSG node, select **OK** and press **Enter**.
32. Enter the **root** password, select **OK** and press **Enter**.
The DSG node is restarted.
33. Login to the DSG Web UI.
34. Click the ⓘ (Help) icon, and then click **About**.
35. Verify that the DSG version is reflected as *DSG 3.1.0.5*.
36. Login to the ESA Web UI using the administrator credentials.
37. Navigate to **Cloud Gateway > Cluster**.
38. Click **Deploy** to push the DSG Ruleset configurations to the DSG nodes.

Note:

The DSG Ruleset will be pushed only if you [Add the DSG node](#) after the installation.

39. Login to the DSG Web UI using the administrator credentials.
40. Navigate to **Logs > Appliance**.
41. Click **Cloud Gateway - Event Logs**, and select **Gateway**.
Verify that the startup logs do not display any errors.
For more information about handling error at startup, refer to the section [Appendix F: Troubleshooting Data Security Gateway \(DSG\)](#).

8.2 Installing the DSG on Cloud Platforms

This section provides information about installing the DSG on cloud platforms.

8.2.1 Installing Data Security Gateway (DSG) on Amazon Web Services (AWS)

This section describes the process for launching a Data Security Gateway (DSG) instance on Amazon Web Services (AWS).

Amazon Web Services (AWS) is a cloud-based computing service, which provides several services such as computing power through Amazon Elastic Compute Cloud (EC2), storage through Amazon Simple Storage Service (S3), and so on. The AWS stores Amazon Machine images (AMIs), which are templates or virtual images containing an operating system, applications, and configuration settings.

8.2.1.1 Prerequisites

This section describes the required prerequisites for launching and installing the DSG on AWS. It also includes the information for the audience, network prerequisites, and hardware and software requirements for the DSG.

Ensure that the following prerequisites are met before launching the DSG on AWS:

- The ESA 9.1.0.5 is installed.

Note:

For more information about installing the ESA 9.1.0.5, refer to the section *Installing Appliance On-Premise* and *Installing Appliances on Cloud Platforms* in the *Protegility Installation Guide 9.1.0.5*.

Caution:

Ensure that the Policy Management (PIM) has been initialized on the ESA. The initialization of the PIM ensures that cryptographic keys for protecting data and the policy repository have been created.

For more information about initializing the PIM, refer to the section *Initializing the Policy Management* in the *Protegility Policy Management Guide 9.1.0.5*.

Caution: Ensure that the **Analytics** is initialized on the ESA. The initialization of the **Analytics** displays the information from the Audit Store.

For more information about initializing the **Analytics**, refer to the section *Initializing the Audit Store Cluster on the ESA* in the *Protegility Installation Guide 9.1.0.5*.

Caution:

Ensure that a Trusted Appliance Cluster (TAC) is created on the ESA.

For more information about creating a TAC on the ESA, refer to the section *Configuring the DSG Cluster*.

- An Amazon account for using AWS is available with the following information:
 - Login URL for the AWS account
 - Authentication credentials for the AWS account

8.2.1.1.1 Audience

This section contains information for stakeholders who are interested in understanding how to create, launch, and install a DSG instance on AWS.

It is recommended that you understand and use Amazon Web Services and its related concepts.

For more information about the Amazon Web Services concepts, refer to the AWS documentation at <https://docs.aws.amazon.com>.

8.2.1.1.2 Hardware Requirements

This section describes the hardware requirements for the DSG.

As the Protegility appliances are hosted and run on AWS, the hardware requirements are dependent on the configurations provided by Amazon.

For reference, the following list describes the minimum hardware requirements for the DSG:

- CPU: 4 Cores
- RAM: 16 GB
- Disk Size: 64 GB
- Network Interfaces: 2

Note: The hardware configuration required might vary based on the actual usage or amount of data and logs expected.

8.2.1.1.3 Network Requirements

This section describes the network requirements for a DSG instance on AWS.

It is recommended that the DSG on AWS must be installed in the Amazon Virtual Private Cloud (VPC) networking environment.

For more information about the Amazon Virtual Private Cloud, refer to the documentation at: <http://docs.aws.amazon.com>.

Note:

Ensure that two Network Interface Cards (NICs) are added during the DSG instance creation on AWS.

For more information about the network interface requirements, refer to the section [Network Planning](#).

The Data Security Gateway must be configured with the following two network interfaces:

- Management Interface - This interface is used for communication between the ESA and the DSG, and accessing the DSG Web UI.
- Service Interface - This interface is used for handling the network traffic traversing through the DSG.

8.2.1.2 Backing Up and Restoring the DSG Instance Snapshot on AWS

It is recommended that a snapshot of the DSG instance is taken such that it can be restored in the event of a failure.

For more information about creating snapshots in AWS, refer to the AWS documentation at <https://docs.aws.amazon.com>.

8.2.1.3 Installing and Launching DSG

The installation of the DSG in AWS involves applying the DSG patch on the ESA and installing the DSG in AWS.

Note: Ensure that the prerequisites for installing the DSG are met.

For more information about prerequisites, refer to the section [Prerequisites](#).

1. Apply the DSG v3.1.0.5 patch on the ESA v9.1.0.5 instance.

For more information about applying the patch on the ESA, refer to the section [Applying the DSG 3.1.0.5 Patch](#).

2. Install and launch the DSG instance created using DSG v3.1.0.5 AMI.

For more information about installing and launching the DSG instance, refer to the section [Installing the DSG on AWS](#).

8.2.1.3.1 Installing the DSG patch on ESA

This section includes the steps to install the DSG v3.1.0.5 patch on the ESA.

You must apply the DSG v3.1.0.5 patch on the ESA v9.1.0.5. This step ensures that the DSG component version on the ESA is updated to v3.1.0.5.

Note: For more information about applying a patch, refer to the section [Extending ESA with DSG Web UI](#).

8.2.1.3.2 Installing the DSG on AWS

This section provides information for the steps required to launch and install the Data Security Gateway (DSG) instance from an AMI provided by Protegility.

Ensure that the installation order provided in the table is followed.

Table 8-3: Order of installation/configuration - AWS

Order of installation	Description	Reference
1	Create and launch the DSG instance	Creating and Launching a DSG Instance from the AMI
2	Finalize the DSG Installation	Finalizing the DSG Installation
3	Configure the Default Gateway for the Management NIC using the DSG CLI Manager	Configuring Default Gateway for Management NIC (ethMNG) using the DSG CLI Manager
4	Configure the Default Gateway for the Service NIC using the DSG CLI Manager	Configuring Default Gateway for Service NIC (ethSRV0) using the DSG CLI Manager
5	Set up ESA Communication	Setting up ESA Communication
6	Forward logs to the Audit Store	Forwarding Logs to the Audit Store
7	Post DSG Installation Steps	Post DSG installation Steps
8	Optional: Install Cloud Utility AWS	Installing the Cloud Utility AWS Tool

8.2.1.3.2.1 Creating and Launching a DSG Instance from the AMI

This section includes the steps to create a DSG instance from the AMI.

Before you begin

Ensure that the DSG AMI is downloaded from the My.Protegility portal to your AWS account.

► To create an instance of DSG:

1. Login to the AWS Management Console.
2. On the **AWS Management Console** screen, click **Services**.
3. Under the **Compute** section, click **EC2**.
4. On the **EC2 Dashboard** screen, under the **Images** area, click **AMIs**.
The AMIs that are accessible to the user account appear in the right pane.
5. Select the DSG AMI, *DSG_PAP-ALL-64_x86-64_AWS_3.1.0.5.x.ami*.

6. Click **Launch** to launch the selected DSG appliance.
7. On the **Choose an Instance** screen, select the required instance type.

Caution: It is recommended that an instance with minimum **4 Core CPU** and **16 GB RAM** configuration is selected. The hardware configuration required might vary based on the actual usage or amount of data and logs expected.

8. Click **Next: Configure Instance Details** to configure the details of the instance. Ensure that you set the other parameters on this screen based on your requirements.

Note: It is recommended that any Protegility appliances on AWS must be installed in the Amazon Virtual Private Cloud (VPC) networking environment.

9. Under the **Network Interfaces** area, click **Add Device** to add a second network interface for your instance.

Note:

For ensuring network security, the DSG isolates the management interface from the service interface by allocating each with a separate network address. Ensure that two NICs are added to the DSG.

▼ Network interfaces ①					
Device	Network Interface	Subnet	Primary IP	Secondary IP addresses	IPv6 IPs
eth0	New network interface ▾	subnet-05031e66 ▾	Auto-assign	Add IP	Add IP
eth1	New network interface ▾	subnet-05031e66 ▾	Auto-assign	Add IP	Add IP

Figure 8-26: Adding the second NIC to the instance

10. Click **Next: Add Storage** to add the storage details of a DSG instance.

Note:

Ensure that the size of the storage is a minimum of **64 GB**.

Note: You can add volumes based on your requirements to handle log storage or policy data storage. You can provision additional storage for the appliance by clicking **Add New Volume**. The **Root** volume type is the default volume for your instance.

11. Click **Next: Add Tags** to create a key-value pair that can help identify the instance when you try to search for it.
12. Click **Next: Configure Security Group** to configure the Security Group-related details.

Note:

You must add the required inbound and outbound ports to the security group.

For more information about the ports that must be configured, refer to the section [Network Planning](#).

13. Click **Review and Launch** to review the details related to the DSG instance.

The **Review and Launch** screen appears. It lists all the details used to configure the DSG instance. You can review the required sections before you launch your instance.

14. Click **Launch** to launch the DSG instance.
15. In the **Key Pair** dialog box, select the required key pair option from the drop-down list.

Note:

It is mandatory to select and download the key pair details.

16. Select the acknowledgment check box.
17. Click **Launch Instances**.
The DSG instance is launched and the **Launch Status** screen appears.
18. In the **Launch Status** screen, click the launched instance link.
The **Instances** screen appears. It lists the DSG instance-related details.

After the instance is created, you must finalize the DSG installation by accessing the instance using the instance IP.

8.2.1.3.2.2 Finalizing the DSG Installation

After the DSG instance is launched, you must complete the process to finalize the DSG installation to rotate the Protegility provided keys and certificates so that these are regenerated as a security best practice.

Before you begin

Caution:

Ensure that the SSH connection is not interrupted during the finalization of the DSG installation. If the SSH connection is interrupted, then the finalization of the DSG installation fails. The process of instance creation and installation of the DSG must be started afresh.

► To finalize the DSG installation:

1. Access the DSG instance IP and provide the downloaded key pair details to use an SSH client.
2. Login using the *local_admin* user for the DSG.
3. Press **Tab** to select **Yes** and press **Enter** to finalize the installation.

The finalize installation confirmation screen appears.



Figure 8-27: Finalize Installation

If you select **No** during finalization, then the DSG installation does not complete.

Perform the following steps to complete the finalization of the DSG installation on the DSG CLI manager.

1. Navigate to **Tools > Finalize Installation**.
2. Follow the *step 4* to *step 6* to complete installing the DSG.
4. Press **Tab** to select **Yes** and press **Enter** to rotate the required keys, certificates, and credentials for the appliance.

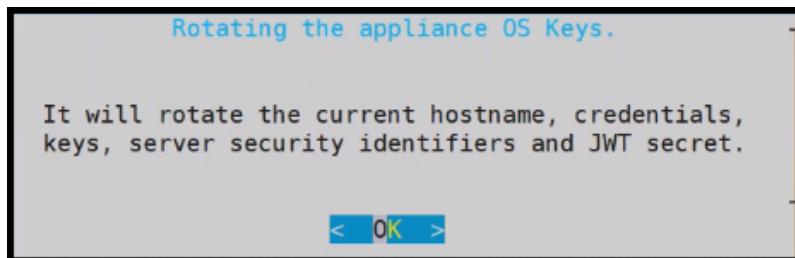


Figure 8-28: Rotating the appliance OS Keys Dialog box

5. Configure the default user passwords, press **Tab** to select **Apply** and press **Enter**.

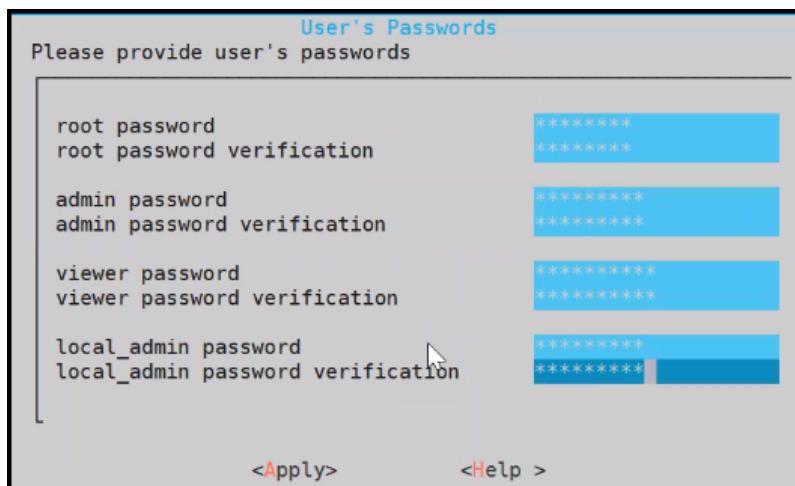


Figure 8-29: User Passwords screen

Note: It is recommended that strong passwords are set for all the users.

For more information about password policies, refer to the section *Strengthening Password Policy* in the *Protegility Appliances Overview Guide 9.1.0.5*.

Ensure that the default passwords are not reused.

6. Press **Tab** to select **Continue** and press **Enter** to complete the finalization of the DSG installation.

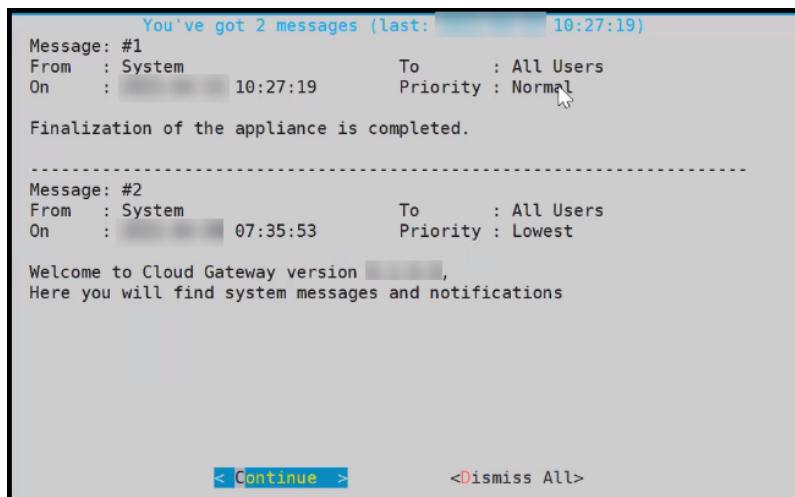


Figure 8-30: Successful DSG finalization

7. Login to the DSG CLI Manager through the *local_admin* user credential.
8. Navigate to **Administration > Reboot and Shutdown**.
9. Select **Reboot** and press **Enter**.
10. Provide a reason for restarting the DSG node, select **OK** and press **Enter**.
11. Enter the *root* password, select **OK** and press **Enter**.
The DSG node is restarted.
12. To access the DSG CLI Manager with *administrator* user credentials, you must login to the DSG Web UI. On the DSG Web UI, navigate to **Settings > Network** and select the **SSH** tab. In the SSH Authentication Configuration section, select **Password + Publickey** as the Authentication type and click **Apply**.

The finalization of the DSG installation completes successfully.

8.2.1.4 Best Practices for Using DSG on AWS

The following table lists some best practices for using Protegility appliances on AWS.

Table 8-4: Best Practices for using Protegility Appliances on AWS

Practice	Description
Force SSH Keys	<p>Configure the appliance to enable SSH keys and disable SSH passwords for all users.</p> <p>If you need to create or join a Trusted Appliance cluster, then ensure that SSH passwords are enabled when you are creating or joining the cluster, and then disabled.</p> <p>For more information about the SSH keys, refer to the section <i>Working with Secure Shell (SSH) Keys</i> in the <i>Protegility Appliances Overview Guide 9.1.0.5</i>.</p>
Enable SSH for the <i>local_admin</i> user account	<p>By default, the <i>local_admin</i> user account does not have SSH access. However, for AWS, it is recommended to provide SSH access to the <i>local_admin</i> user account so that the appliance can be accessed in case of any contingency.</p> <p>It is recommended to enable SSH access and SSH keys, and disable SSH passwords to the <i>local_admin</i> user account on the appliance.</p>

Practice	Description
	For more information about the SSH keys, refer to the section <i>Working with Secure Shell (SSH) Keys</i> in the <i>Protegility Appliances Overview Guide 9.1.0.5</i> .

8.2.2 Installing the Data Security Gateway (DSG) on Microsoft Azure

This section provides information on launching a Data Security Gateway (DSG) virtual machine (VM) on the Microsoft Azure platform.

The Microsoft Azure platform is a set of cloud-based computing services, which include computing services, virtual machines, data storage, analytics, networking services, and so on.

8.2.2.1 Prerequisites

This section describes the required prerequisites for launching the DSG on Azure. It also includes the details for the network prerequisites and hardware requirements for the DSG.

Ensure that the following prerequisites are met before launching the DSG on Azure:

- The ESA 9.1.0.5 is installed.

Note:

For more information about installing the ESA 9.1.0.5, refer to the section *Installing Appliance On-Premise* and *Installing Appliances on Cloud Platforms* in the *Protegility Installation Guide 9.1.0.5*.

Caution:

Ensure that the Policy Management (PIM) has been initialized on the ESA. The initialization of the PIM ensures that cryptographic keys for protecting data and the policy repository have been created.

For more information about initializing the PIM, refer to the section *Initializing the Policy Management* in the *Protegility Policy Management Guide 9.1.0.5*.

Caution: Ensure that the **Analytics** is initialized on the ESA. The initialization of the **Analytics** displays the information from the Audit Store.

For more information about initializing the **Analytics**, refer to the section *Initializing the Audit Store Cluster on the ESA* in the *Protegility Installation Guide 9.1.0.5*.

Caution:

Ensure that a Trusted Appliance Cluster (TAC) is created on the ESA.

For more information about creating a TAC on the ESA, refer to the section *Configuring the DSG Cluster*.

- An Azure account is available with the following information:
 - Sign in URL for the Azure account
 - Authentication credentials for the Azure account
- Ensure that the DSG BLOB is available in the storage account that will be selected to create the disk and the VM.



8.2.2.1.1 Audience

This section contains information for stakeholders who are interested in understanding how to create, launch, and install a DSG instance on Azure.

It is recommended that you possess working knowledge of the Azure Platform and knowledge of related concepts.

For more information about Azure concepts, refer to the Azure documentation at: <https://docs.microsoft.com/en-us/azure/>

8.2.2.1.2 Hardware Requirements

This section describes the hardware and software requirements for the DSG.

As the DSG is hosted and run on Azure, the hardware requirements are dependent on the configurations provided by Azure.

For reference, the following list describes the minimum hardware requirements for the DSG:

- CPU: 4 Cores
- RAM: 16 GB
- Disk Size: 64 GB
- Network Interfaces: 2

Note: The hardware configuration required might vary based on the actual usage or amount of data and logs expected.

8.2.2.1.3 Network Requirements

This section explains the network requirements for the DSG in Azure.

It is recommended that the DSG on Azure is provided with the Azure Virtual Network environment.

For more information about the Azure Virtual Network, refer to the Azure documentation at <https://docs.microsoft.com/en-us/azure/>.

Caution:

Ensure that two Network Interface Cards (NICs) are added during the DSG instance creation on Azure.

For more information about the network interface requirements, refer to the section [Network Planning](#).

The Data Security Gateway must be configured with the following two network interfaces:

- Management Interface - This interface is used for communication between the ESA and the DSG, and accessing the DSG Web UI.
- Service Interface - This interface is used for handling the network traffic traversing through the DSG.

8.2.2.2 Backing Up and Restoring the DSG Instance Snapshot on Azure

This section provides information on backing up the DSG instance on Azure. It is recommended that a snapshot of the DSG instance is taken such that it can be restored in the event of a failure.

You can create a backup for a virtual machine by using either of the following two Linux VM Agent methods:

1. Creating snapshots of the disk
2. Using Recovery Services Vaults

For more information about enabling backup using the Linux VM Agent methods, refer to the section [Working with Linux VM Agent](#).

For more information about creating a backup for a virtual machine, refer to the Azure documentation at <https://docs.microsoft.com/en-us/azure>.

Note: If you are using the *Create a new VM* restore option provided by Microsoft Azure to restore a DSG VM instance, then the Service IP of the restored DSG VM instance must be updated. The Service IP of the DSG VM can be updated by using the steps provided in the section [Configuring the Second Network Interface](#).

8.2.2.3 Installing and Launching DSG

The installation of the DSG in Azure involves applying the DSG patch on the ESA and installing the DSG in Azure.

Note: Ensure that the prerequisites for installing the DSG are met.

For more information about prerequisites, refer to the section [Prerequisites](#).

1. Apply the DSG v3.1.0.5 patch on the ESA v9.1.0.5 instance.

For more information about applying the patch on the ESA, refer to the section [Applying the DSG 3.1.0.5 Patch](#).

2. Install and launch the DSG instance created using DSG v3.1.0.5 image.

For more information about installing and launching the DSG instance, refer to the section [Installing the DSG on AWS](#).

8.2.2.3.1 Installing the DSG patch on ESA

This section includes the steps to install the DSG v3.1.0.5 patch on the ESA.

You must apply the DSG v3.1.0.5 patch on the ESA v9.1.0.5. This step ensures that the DSG component version on the ESA is updated to v3.1.0.5.

Note: For more information about applying a patch, refer to the section [Extending ESA with DSG Web UI](#).

8.2.2.3.2 Installing and Launching DSG on Azure

This section provides information for the steps required to launch and install the Data Security Gateway (DSG) instance from a BLOB provided by Protegility.

Ensure that the installation order provided in the table is followed.

Table 8-5: Order of installation/configuration - Azure

Order of installation	Description	Reference
1	Create a Disk from a BLOB	Creating Image from the DSG BLOB
2	Create a VM from a Disk	Creating a VM from the Image
3	Adding the Second Network Interface	Adding the Second Network Interface
4	Finalize the DSG Installation	Finalizing the DSG Installation
5	Configuring the Second Network Interface	Configuring the Second Network Interface
6	Configure the Default Gateway for the Management NIC using the DSG CLI Manager	Configuring Default Gateway for Management NIC (ethMNG) using the DSG CLI Manager
7	Configure the Default Gateway for the Service NIC using DSG CLI Manager	Configuring Default Gateway for Service NIC (ethSRV0) using the DSG CLI Manager



Order of installation	Description	Reference
8	Set Up ESA Communication	Setting up ESA Communication
9	Forward Logs to the Audit Store	Forwarding Logs to the Audit Store
10	Post DSG Installation Steps	Post DSG installation Steps
11	Optional: Install Cloud Utility AWS	Installing the Cloud Utility AWS Tool

8.2.2.3.2.1 Creating Image from the DSG BLOB

This section explains how to create an image from the DSG BLOB.

Before you begin

Ensure that the DSG BLOB is downloaded from the My.Protegility portal to your Azure storage account that will be selected to create the image.

► To create an image from the BLOB:

1. Log in to the Azure portal.
2. Select **Images** and click **Create**.
3. Enter the details in the **Resource Group**, **Name**, and **Region** text boxes.
4. In the **OS disk** option, select **Linux**.
5. In the **VM generation** option, select **Gen 1**.
6. In the **Storage blob** drop-down list, select the Protegility Azure BLOB.
7. Enter the appropriate information in the required fields and click **Review + create**.
The image is created from the BLOB.

8.2.2.3.2.2 Creating a VM from the Image

This section describes the steps to create a VM from an image.

After obtaining the image, you can create a VM from it. For more information about creating a VM from the image, refer to the following link.

<https://docs.microsoft.com/en-us/azure/virtual-machines/linux/quick-create-portal#create-virtual-machine>

► To create a VM:

1. Login in to the Azure homepage.
2. Click **Images**.
The list of all the images appear.
3. Select the required image.
4. Click **Create VM**.
5. Enter details in the required fields.
6. Select **SSH public key** in the **Authentication type** option.

Note:



As a security measure for the appliances, it is recommended to not use the **Password based mechanism** as an authentication type.

7. In the **Username** text box, enter the name of a user.

Note:

This user is added as an OS level user in the appliance. Ensure that the following usernames are not provided in the **Username** text box:

- Appliance OS users
- Appliance LDAP users

8. Select the required SSH public key source.
 9. Enter the required information in the *Disks*, *Networking*, *Management*, and *Tags* sections.
 10. Click **Review + Create**.
- The VM is created from the image.
11. After the VM is created, you can access the appliance from the CLI Manager or Web UI.

Note:

The OS user that is created in step 7 does not have SSH access to the appliance. If you want to provide SSH access to this user, login to the appliance as another administrative user and [toggle SSH access](#). In addition, update the user to permit [Linux shell access](#) (`/bin/sh`).

8.2.2.3.2.3 Adding the Second Network Interface

For ensuring network security, the DSG isolates the management interface from the service interface by allocating each with a separate network address. Ensure that two NICs are added to the DSG. This section explains the steps to add a second network interface to the DSG appliance after a DSG VM is created.

► To add a second network interface to the DSG:

1. On the Azure Portal Dashboard, click **Virtual Machines**.
2. Select the DSG VM that you created.

The DSG VM details appear in the *Virtual Machine* screen.
3. On the *Virtual Machine* screen, click **Overview**.
4. Click **Stop** to power off the VM.
5. Create the second network interface for the DSG VM.
6. Navigate to the *Virtual Machine* screen, select the DSG VM that you created, and click **Networking** under the *Settings* area.
7. Click **Attach network interface**.
8. Select the network interface that you created in [step 5](#), and click **OK**.

Note:

The second network interface is added to the VM. You can view two tabs that represent NICs for the management and service interfaces.

9. Click **Start** to power on the VM.

The second network interface is added to the DSG node.

8.2.2.3.2.4 Finalizing the DSG Installation

After the DSG instance is launched, you must complete the process to finalize the DSG installation to rotate the Protegility provided keys and certificates so that these are regenerated as a security best practice.

Before you begin

Caution:

It is recommended to finalize the installation of the DSG instance using the *Serial Console* provided by Azure. Do not finalize the installation of the DSG instance using the SSH connection.

► To finalize the DSG installation:

1. Sign in to the Azure homepage.
2. On the left pane, click **Virtual machines**.
The *Virtual machine* screen appears.
3. Select the required virtual machine and click **Serial console**.
The DSG CLI manager screen appears.
4. Login to the DSG CLI Manager using the administrator credentials and press **ENTER**.

Note:

The credentials for logging in to the DSG are provided in the DSG 3.1.0.5 readme.

5. Press **Tab** to select **Yes** and press **Enter** to finalize the installation.

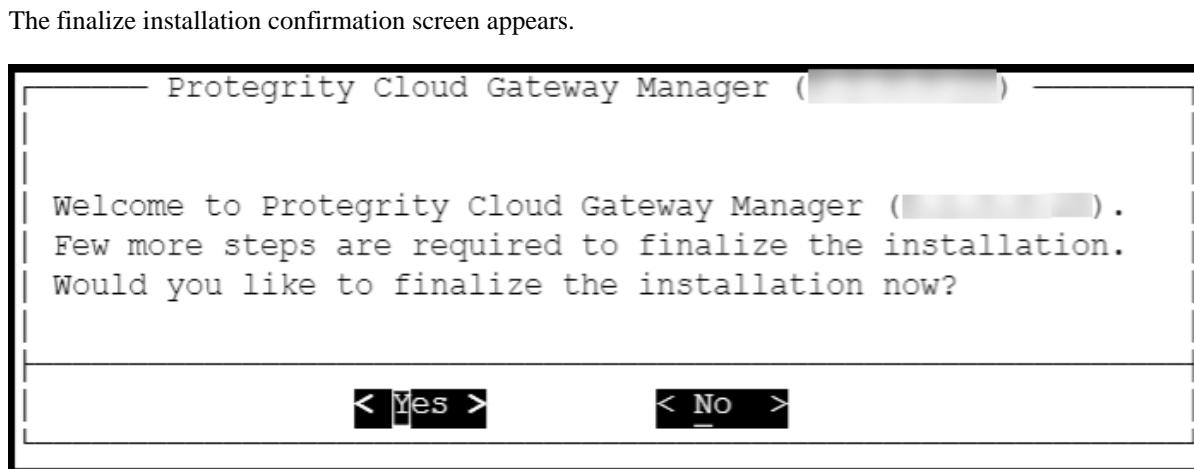
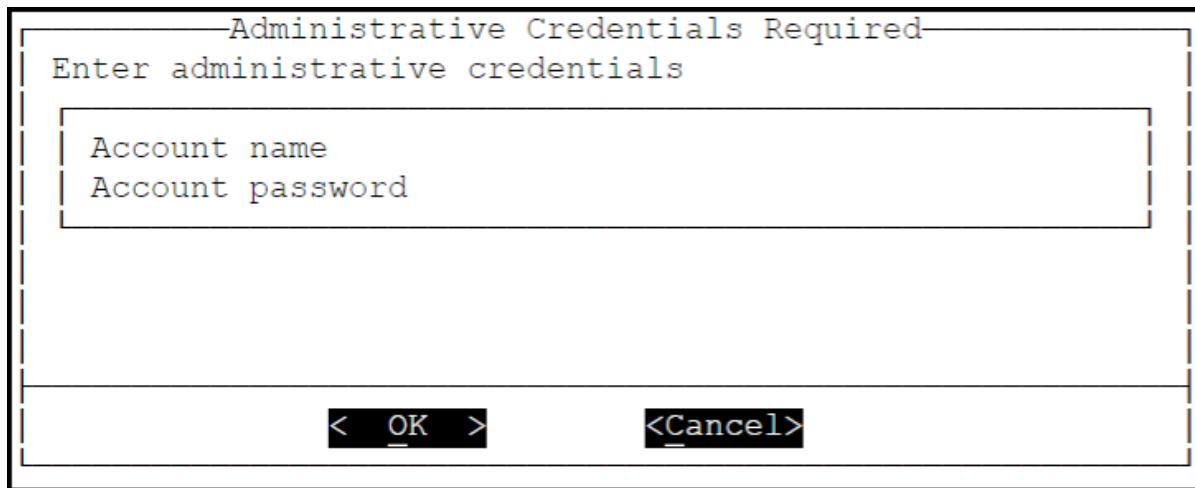


Figure 8-31: Finalize Installation

If you select **No** during finalization, then the DSG installation does not complete.

Perform the following steps to complete the finalization of the DSG installation on the DSG CLI manager.

1. Navigate to **Tools > Finalize Installation**.
2. Follow the *step 6* to *step 9* to complete installing the DSG.
6. Enter the administrator credentials for the DSG instance, press **Tab** to select **OK** and press **Enter**.



Note:

The administrator credentials for logging in to the DSG are provided in the DSG 3.1.0.5 readme.

Figure 8-32: Entering DSG administrator credentials

7. Press **Tab** to select **Yes** and press **Enter** to rotate the required keys, certificates, and credentials for the appliance.

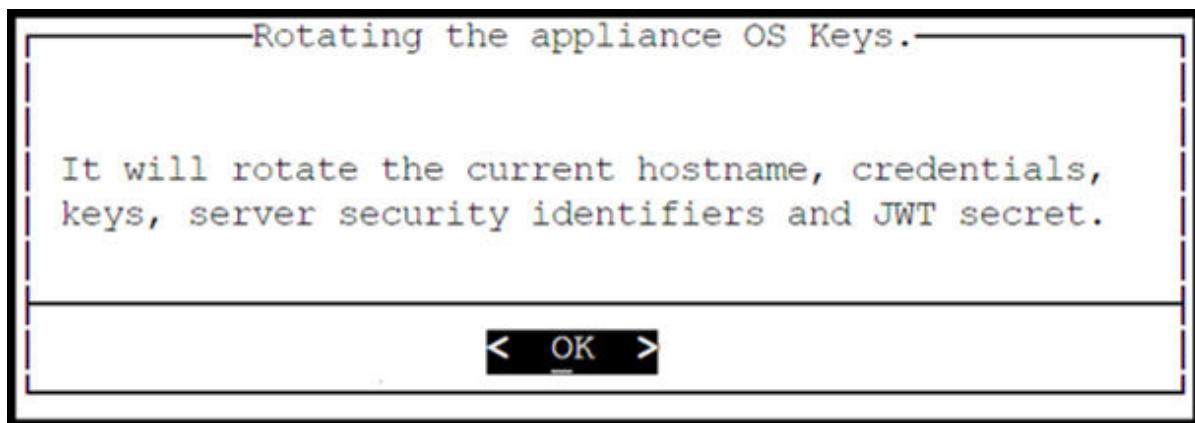


Figure 8-33: Appliance OS Keys Rotation Dialog box

8. Configure the default user's passwords, press **Tab** to select **Apply** and press **Enter** to continue.

User's Passwords

Please provide user's passwords

root password	[REDACTED] *****
root password verification	[REDACTED] *****
admin password	[REDACTED] *****
admin password verification	[REDACTED] *****
viewer password	[REDACTED] *****
viewer password verification	[REDACTED] *****
local_admin password	[REDACTED] *****
local_admin password verification	[REDACTED] *****

<Apply> **<Help >**

Figure 8-34: User's Passwords screen

Note: It is recommended that strong passwords are set for all the users.

For more information about password policies, refer to the section *Strengthening Password Policy* in the *Protegility Appliances Overview Guide 9.1.0.5*.

Ensure that the default passwords are not reused.

9. Press **Tab** to select **Continue** and press **Enter** to complete the finalization of the DSG installation.

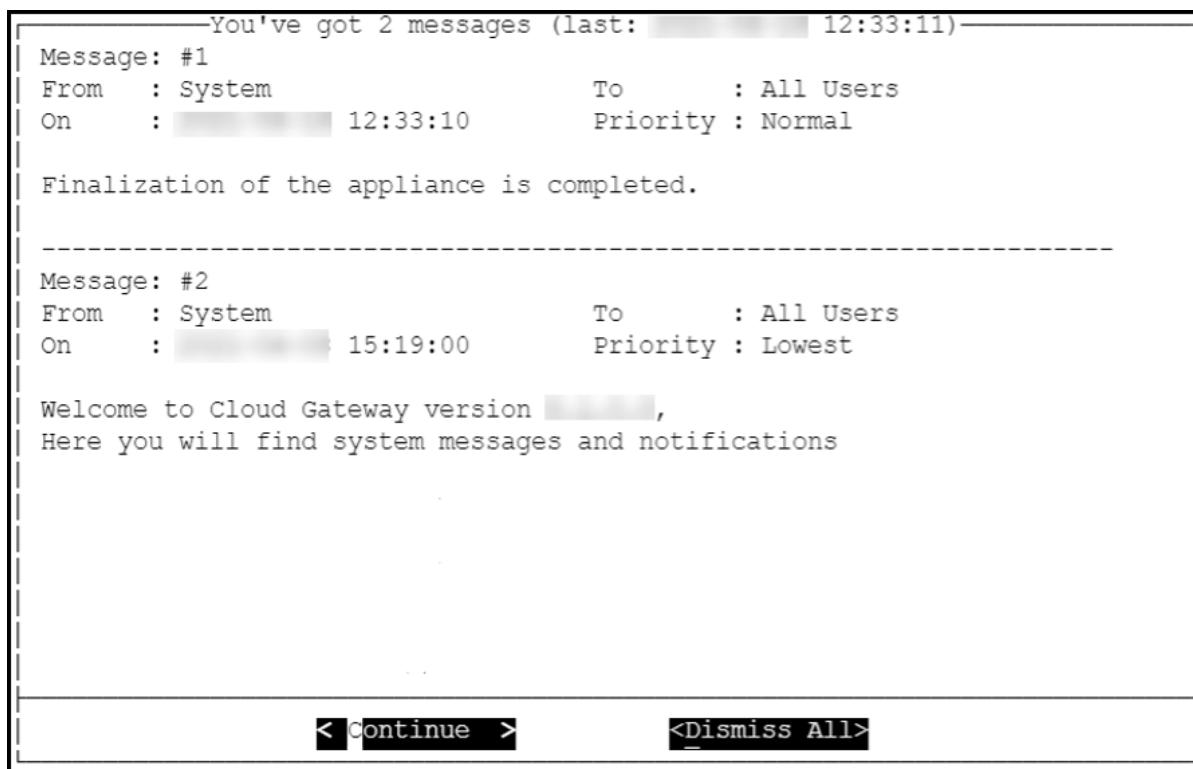


Figure 8-35: Successful DSG finalization

10. Login to the DSG CLI Manager.
 11. Navigate to **Administration > Reboot and Shutdown**.
 12. Select **Reboot** and press **Enter**.
 13. Provide a reason for restarting the DSG node, select **OK** and press **Enter**.
 14. Enter the **root** password, select **OK** and press **Enter**.
- The DSG node is restarted.

A part of finalization of the DSG installation is completed successfully. For the next part of the installation, the *second network interface* must be configured before you can use the DSG instance.

8.2.2.3.2.5 Configuring the Second Network Interface

This section explains the steps to configure a second network interface on the DSG after finalizing the DSG installation.

► To configure the second network interface on the DSG:

1. On the Azure Portal Dashboard, click **Virtual Machines**.
2. Navigate to the *Virtual Machine* screen, and select the DSG VM instance that you created earlier.
3. Click **Overview**.
4. Click **Serial Console** to access the DSG instance.
5. Login to the DSG instance using the administrator credentials.
6. Navigate to **Networking > Network Settings**.

The *Network Configuration Information* screen appears.

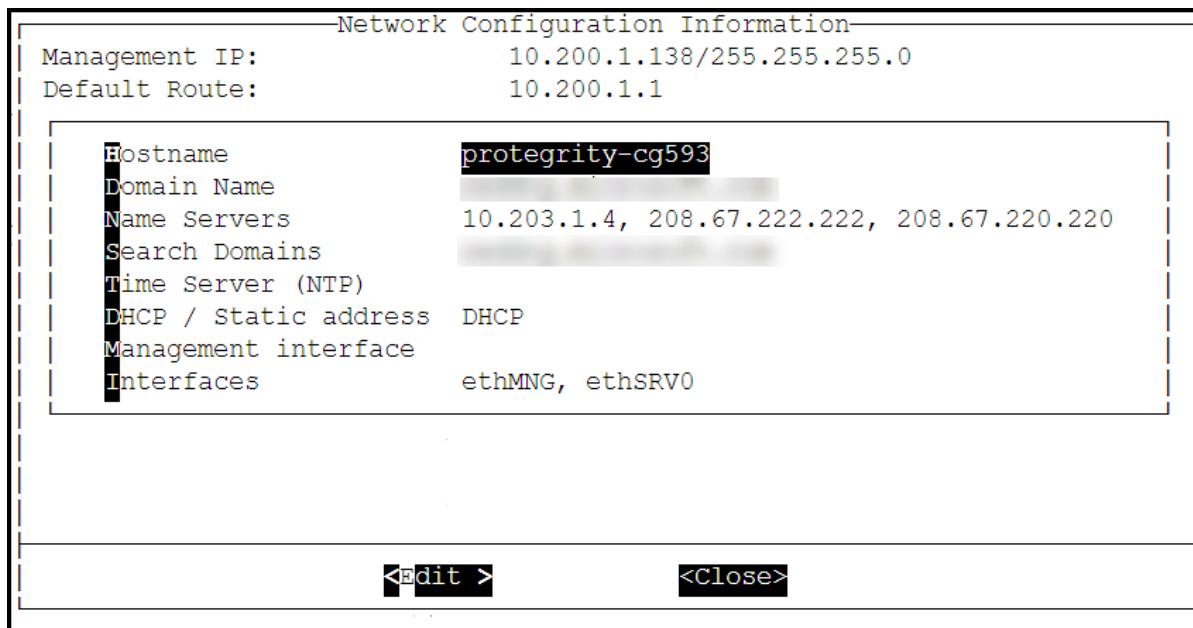


Figure 8-36: Network Configuration Information screen

7. Select **Interfaces** and press **Edit**.
8. Select the **ethSRV0** interface and proceed by pressing **Tab** to select **Edit**.

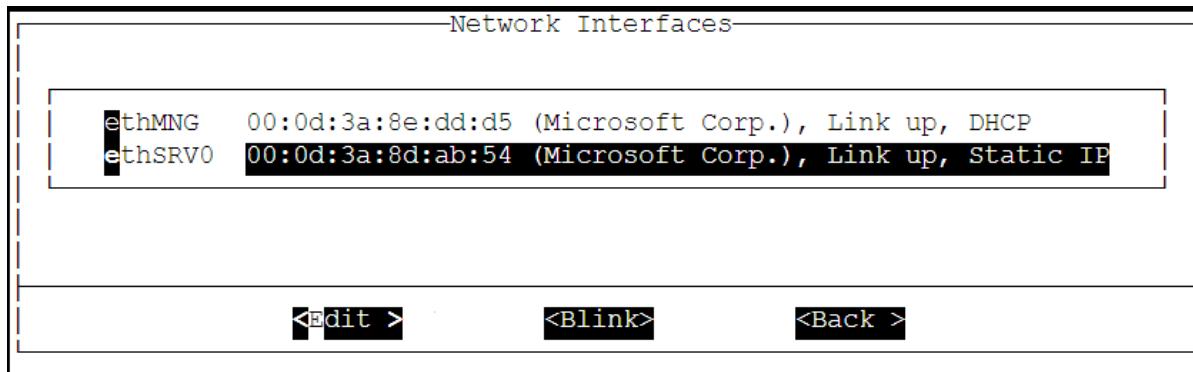


Figure 8-37: Updating the ethSRV0 network interface

9. Select either **DHCP** or **Static** for the **ethSRV0** interface.

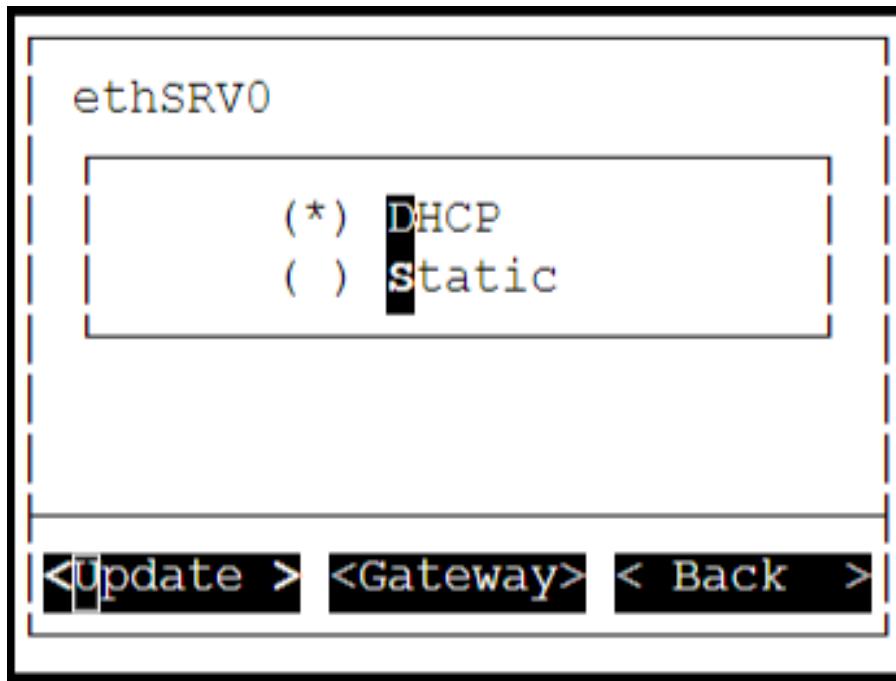


Figure 8-38: Updating network settings

- a. If the DHCP server is not configured, then select **Static**, and proceed by pressing **Tab** to select **Update** for updating the network information manually.

The *Interface Settings* screen appears.

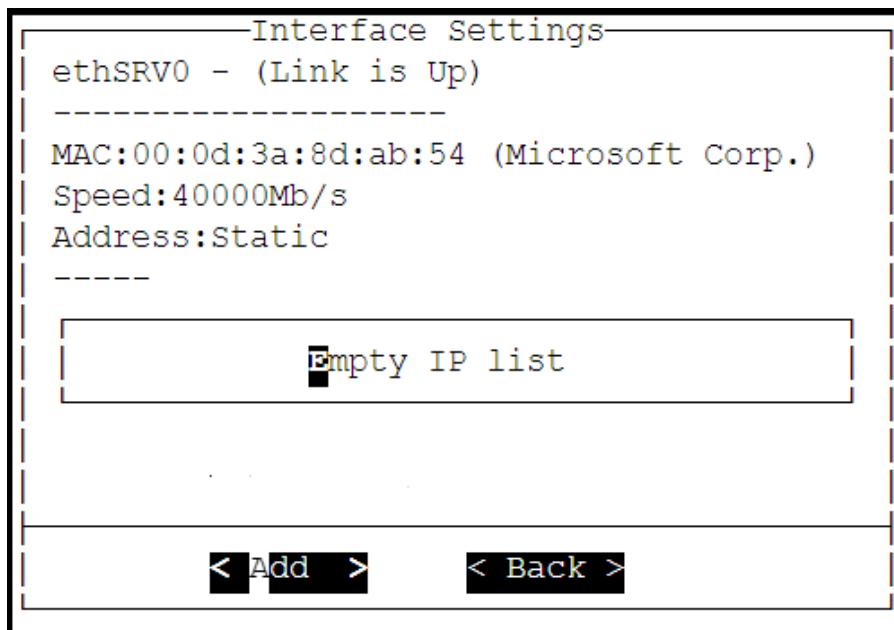


Figure 8-39: Interface Settings screen

10. On the *Interface Settings* screen, press **Tab** and select **Add** to enter the IP Address and Netmask for the *ethSRV0* interface.

The *Network Settings* screen appears.

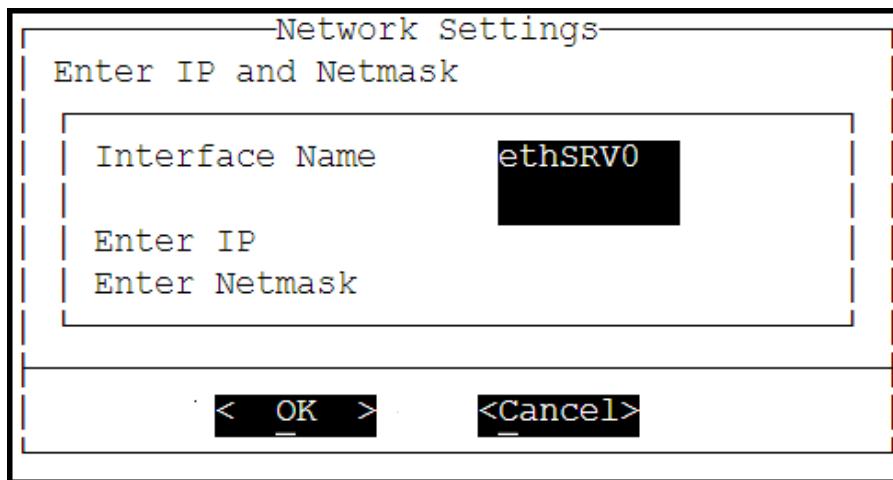


Figure 8-40: Network Settings screen

- On the *Network Settings* screen, enter the **IP Address** and the **Netmask** of the *ethSRV0* interface and proceed by pressing **Tab** and select **OK**.

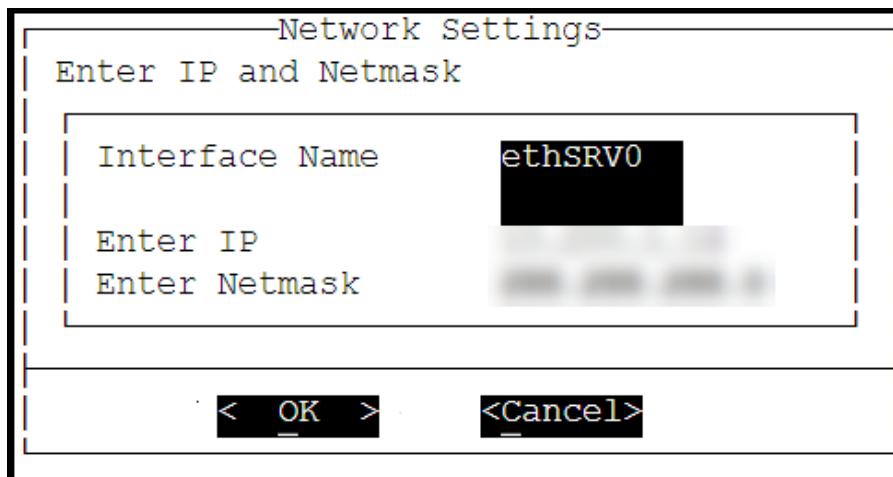


Figure 8-41: Adding network settings

The second network interface, *ethSRV0*, is configured on the DSG node.

8.2.2.4 Azure Cloud Utility

The Azure Cloud Utility is a DSG appliance component that is used for supporting features specific to Azure Cloud Platform, which are, Azure Accelerated Networking and Azure Linux VM agent. When you install the DSG from an Azure v3.1.0.5 BLOB, the Azure Cloud Utility is installed automatically on the DSG.

Caution:

While you are using the Azure Accelerated Networking or Azure Linux VM agent, ensure that the Azure Cloud Utility is not uninstalled.

8.2.2.4.1 Working with Accelerated Networking

The Accelerated Networking is a feature provided by Microsoft Azure, which allows the DSG appliance to handle increasing loads. The advantages offered with Accelerated Networking include reduced latency, reduced jitter, and improved CPU

utilization. The following observations are applicable to the Accelerated Networking feature when it is enabled or not enabled in the VM:

- When this feature is enabled in the VM, the network traffic is routed to the VM Network Interface (NIC), and it is then forwarded to the VM. This helps to improve the networking performance as the traffic bypasses the virtual switch.
- When this feature is not enabled in the VM, the networking traffic coming in and out of the VM traverses through the host and the virtual switch.

The DSG is configured with two network interfaces, Management Interface and Service Interface, where the Management Interface is used for communication between the ESA and the DSG, and accessing the DSG Web UI. The Service Interface is used for handling the network traffic traversing through the DSG.

Note: For more information about an overview and how to configure Azure Accelerated Networking, refer to the Azure documentation at <https://docs.microsoft.com/en-us/azure>.

Note:

It is recommended to configure the Accelerated Networking feature after the DSG is installed on Azure VM instance as it improves the networking performance. As the network traffic traverses through the Service Interface on the DSG, it is recommended to enable the Accelerated Networking feature on the Service Interface.

8.2.2.4.2 Working with Linux VM Agent

The Microsoft Azure Linux Agent (*waagent*) is a VM extension provided by Microsoft Azure that manages image provisioning, networking, kernel, integrating third-party softwares on VMs, and so on.

Note: For more information about the Linux VM agent, refer to the Azure documentation at <https://docs.microsoft.com/en-us/azure>.

For the DSG, the Linux VM agent is used for enabling backup and restore using either of the following two methods:

- *Recovery Services Vaults*
- *Creating Images of an Instance*

The *waagent* extension is registered in the *.vhd* file that is provided by Protegility. To use the Linux VM agent feature, you must create an image from the *.vhd* file provided by Protegility.

Note: For more information about creating an image, refer to the section *Creating an Instance from a Custom Virtual Machine Image* in the [Protegility Appliances Overview Guide 9.1.0.5](#).

8.2.3 Installing the Data Security Gateway (DSG) on Google Cloud Platform (GCP)

This section describes the process for launching a Data Security Gateway (DSG) instance on Google Cloud Platform (GCP).

Google Cloud Platform (GCP) is a set of cloud computing services provided by Google, and offers services, such as compute, storage, and networking.

8.2.3.1 Prerequisites

This section describes the required prerequisites for launching the DSG on GCP. It also includes the information for the audience and the network prerequisites for the DSG.

Ensure that the following prerequisites are met before launching the DSG on GCP:

- The ESA 9.1.0.5 is installed.

Note:

For more information about installing the ESA 9.1.0.5, refer to the section *Installing Appliance On-Premise* and *Installing Appliances on Cloud Platforms* in the *Protegility Installation Guide 9.1.0.5*.

Caution:

Ensure that the Policy Management (PIM) has been initialized on the ESA. The initialization of the PIM ensures that cryptographic keys for protecting data and the policy repository have been created.

For more information about initializing the PIM, refer to the section *Initializing the Policy Management* in the *Protegility Policy Management Guide 9.1.0.5*.

Caution: Ensure that the **Analytics** is initialized on the ESA. The initialization of the **Analytics** displays the information from the Audit Store.

For more information about initializing the **Analytics**, refer to the section *Initializing the Audit Store Cluster on the ESA* in the *Protegility Installation Guide 9.1.0.5*

Caution:

Ensure that a Trusted Appliance Cluster (TAC) is created on the ESA.

For more information about creating a TAC on the ESA, refer to the section *Configuring the DSG Cluster*.

- A GCP account is available with the following information:
 - Login URL for the GCP account
 - Authentication credentials for the GCP account

8.2.3.1.1 Audience

This section contains information for stakeholders who are interested in deploying a DSG instance on GCP.

It is recommended that you understand and use the Google Cloud Platform before proceeding further.

For more information about the Google Cloud Platform, refer to the <https://cloud.google.com/docs>.

8.2.3.1.2 Hardware Requirements

This section describes the hardware and software requirements for the DSG.

As the DSG is hosted and run on GCP, the hardware requirements are dependent on the configurations provided by Google.

The following list describes the minimum required configuration for launching the DSG image on the GCP:

- CPU: 4 Cores
- RAM: 16 GB
- Disk Size: 64 GB



- Network Interfaces: 2

Note: The hardware configuration required might vary based on the actual usage or amount of data and logs expected.

8.2.3.1.3 Network Requirements

This section explains the network requirements for the DSG on GCP.

It is recommended that the DSG on GCP must be installed in the GCP Virtual Private Cloud (VPC) networking environment.

For more information about the GCP Virtual Private Cloud, refer to the documentation at: <https://cloud.google.com/vpc/docs>

Note:

Ensure that two Network Interface Cards (NICs) are added during the DSG instance creation on GCP.

For more information about the network interface requirements, refer to the section [Network Planning](#).

The Data Security Gateway must be configured with the following two network interfaces:

- Management Interface - This interface is used for communication between the ESA and the DSG, and accessing the DSG Web UI.
- Service Interface - This interface is used for handling the network traffic traversing through the DSG.

8.2.3.2 Backing Up and Restoring the DSG Instance Snapshot on GCP

It is recommended that a snapshot of the DSG instance is taken such that it can be restored in the event of a failure.

For more information about creating snapshots in GCP, refer to the GCP documentation at <https://cloud.google.com/docs>.

8.2.3.3 Installing and Launching DSG

The installation of the DSG in GCP involves applying the DSG patch on the ESA and installing the DSG on GCP.

Note: Ensure that the prerequisites for installing the DSG are met.

For more information about prerequisites, refer to [Prerequisites](#).

1. Apply the DSG v3.1.0.5 patch on the ESA v9.1.0.5 instance.

For more information about applying the patch on the ESA, refer to the section [Applying the DSG 3.1.0.4 Patch](#).

2. Install and launch the DSG instance created using DSG v3.1.0.5 image.

For more information about installing and launching the DSG instance, refer to the section [Installing the DSG on AWS](#).

8.2.3.3.1 Installing the DSG patch on ESA

This section includes the steps to install the DSG v3.1.0.5 patch on the ESA.

You must apply the DSG v3.1.0.5 patch on the ESA v9.1.0.5. This step ensures that the DSG component version on the ESA is updated to v3.1.0.5.

Note: For more information about applying a patch, refer to the section [Extending ESA with DSG Web UI](#).

8.2.3.3.2 Installing the DSG on GCP

This section provides information for the steps required to launch and install the Data Security Gateway (DSG) instance from an image provided by Protegility.

Ensure that the installation order provided in the table is followed.

Table 8-6: Order of installation/configuration - GCP

Order of installation	Description	Reference
1	Create and launch the DSG instance	Creating a VM Instance from an Image
2	Finalize the DSG Installation	Finalizing the DSG Installation
3	Configure the Default Gateway for the Management NIC using the DSG CLI Manager	Configuring Default Gateway for Management NIC (ethMNG) using the DSG CLI Manager
4	Configure the Default Gateway for the Service NIC using the DSG CLI Manager	Configuring Default Gateway for Service NIC (ethSRV0) using the DSG CLI Manager
5	Set up ESA Communication	Setting up ESA Communication
6	Forward logs to the Audit Store	Forwarding Logs to the Audit Store
7	Post DSG Installation Steps	Post DSG installation Steps
8	Optional: Install Cloud Utility AWS	Installing the Cloud Utility AWS Tool

8.2.3.3.2.1 Creating a VM Instance from an Image

This section describes how to create a VM instance from a DSG image.

Before you begin

Ensure that the DSG image is downloaded from the My.Protegility portal to your GCP account.

► To create a VM instance from an image:

1. Login to the GCP console.
2. Under the **Compute** section, click **Compute Engine > VM instances**.
3. On the **VM instances** screen, click **CREATE INSTANCE**.
The **Create an instance** screen appears.
4. On the **Create an instance** screen, select the configurations as per your requirements. Some of the configurations on this screen must be set as provided in the sub steps so that the DSG can be installed successfully.
 - a. Under **Machine Configuration**, click the **Serial** and the **Machine type** drop down list and select the required configuration.

Caution: It is recommended that an instance with minimum **4 Core CPU** and **16 GB RAM** configuration is selected. The instance type listed is the minimum hardware configuration.

The hardware configuration required might vary based on the actual usage or amount of data and logs expected.

- b. Under **Boot disk**, click **Change**.
 1. Click the **Custom images** tab, and click the DSG image, *dsg-pap-all-64-x86-64-gcp-3-1-0-5-x*, in the **image** drop down list.
 2. Select the required boot disk type and set the value for the **Size(GB)** option as **64** and then click **Select**.



- c. Under **Firewall**, ensure that the **Allow HTTP traffic** and **Allow HTTPS traffic** check boxes are selected.
- d. Click the **Networking** tab and add two NICs.

Note:

For ensuring network security, the DSG isolates the management interface from the service interface by allocating each with a separate network address. Ensure that two NICs are added to the DSG.

5. Click **Create** to create the VM instance.

After the instance is created, a notification stating that the VM instance has been created appears in the **Notifications** tab.

6. On the **VM instances** screen, search or enter the name of the VM instance.
7. Click the VM that you created.

The **VM instance details** screen appears.

Note:

Ensure that you validate the details, such as, Machine Type, Boot disk, Firewall, and the Network Interfaces on the **VM instance details** screen.

8.2.3.3.2.2 Finalizing the DSG Installation

After the DSG instance is launched, you must complete the process to finalize DSG installation to rotate the Protegility provided keys and certificates so that these are regenerated as a security best practice.

Before you begin

Caution:

It is recommended to finalize the installation of the DSG instance using the *Serial Console* provided by GCP. Do not finalize the installation of the DSG instance using the SSH connection.

► To finalize the DSG installation:

1. Login to the GCP console.
2. Under the **Compute** section, click **Compute Engine > VM instances**.
3. On the **VM instances** screen, search or enter the name of the VM instance.
4. Click **Connect to serial console** to access the DSG instance.
5. Login using the administrator credentials for the DSG.

Note:

The credentials for logging in to the DSG are provided in the DSG 3.1.0.3 readme.

6. Press **Tab** to select **Yes** and press **Enter** to finalize the installation.

The finalize installation confirmation screen appears.



Figure 8-42: Finalize the Installation

If you select **No** during finalization, then the DSG installation does not complete.

Perform the following steps to complete the finalization of the DSG installation on the DSG CLI manager.

1. Navigate to **Tools > Finalize Installation**.
2. Follow the *step 7* to *step 10* to complete installing the DSG.
7. Enter the administrator credentials for the DSG instance, press **Tab** to select **OK** and press **Enter**.

Note:

The credentials for logging in to the DSG are provided in the DSG 3.1.0.5 readme.

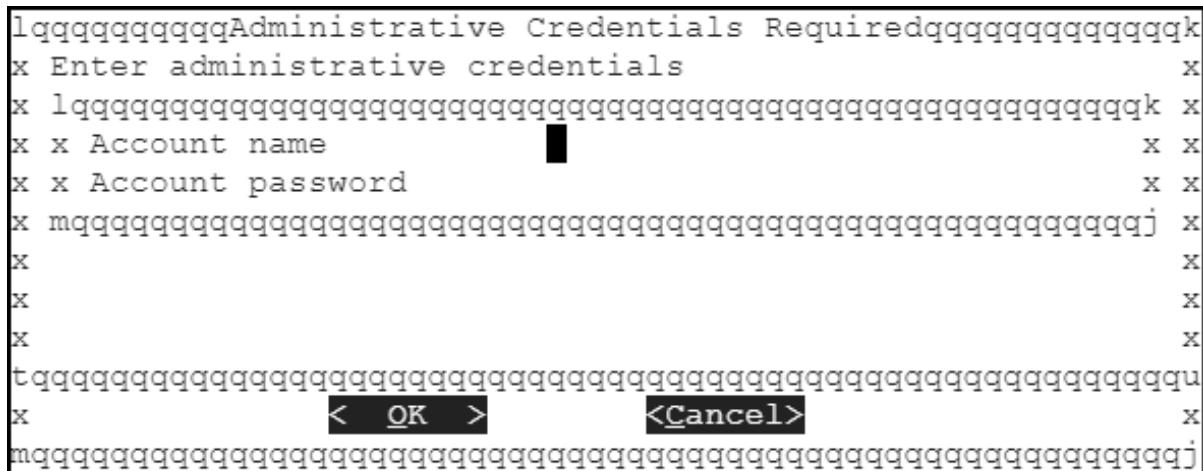


Figure 8-43: Entering DSG administrator credentials

8. Press **Tab** to select **Yes** and press **Enter** to rotate the required keys, certificates, and credentials for the appliance.

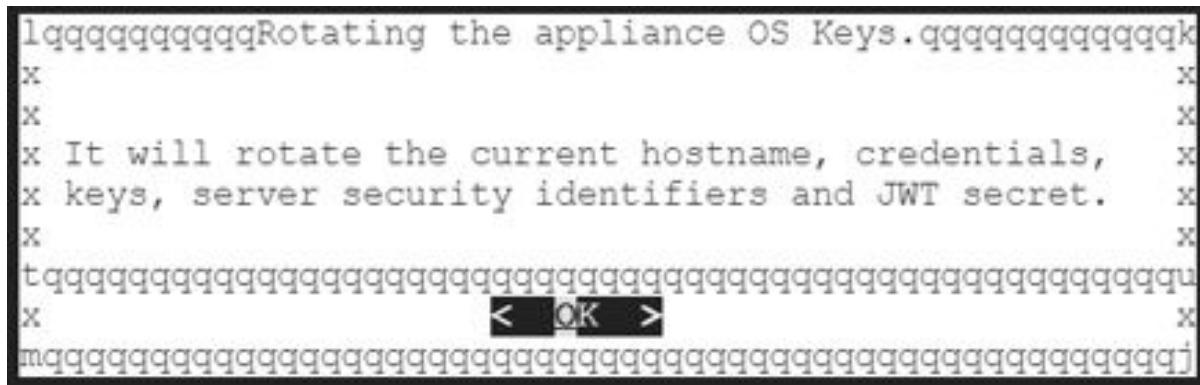


Figure 8-44: Appliance OS Keys Rotation Dialog box

9. Configure the default user's passwords, press **Tab** to select **Apply** and press **Enter** to continue.



Figure 8-45: User's Passwords screen

Note: It is recommended that strong passwords are set for all the users.

For more information about password policies, refer to the section *Strengthening Password Policy* in the *Protegility Appliances Overview Guide 9.1.0.5*.

Ensure that the default passwords are not reused.

10. Press **Tab** to select **Continue** and press **Enter** to complete the finalization of the DSG installation.

Figure 8-46: Successful DSG finalization

11. Login to the DSG CLI Manager.
 12. Navigate to **Administration > Reboot and Shutdown**.
 13. Select **Reboot** and press **Enter**.
 14. Provide a reason for restarting the DSG node, select **OK** and press **Enter**.
 15. Enter the **root** password, select **OK** and press **Enter**.
The DSG node is restarted.

The finalization of the DSG installation completes successfully.

8.2.4 Installing the Cloud Utility AWS Tool

This section provides the steps to install the Cloud Utility AWS Tool. The cloud utility AWS tool is available for all installation on all the cloud platforms supported by Protegility.

AWS offers a variety of cloud-based products for computing, storage, analytics, networking, and management. Using the Cloud Utility AWS Tool, services, such as, CloudWatch and AWS CLI are leveraged by the Protegity appliances. The Cloud Utility AWS Tool version supported by the DSG 3.1.0.5 is *Cloud Utility AWS v2.1.3*.

For more information about using the Cloud Utility AWS Tool, refer to the section *Working with Cloud-based Applications* in the [*Protegity Appliances Overview Guide 9.1.0.5*](#).

Before you begin

- Ensure that you have installed the Cloud Utility AWS Tool on the ESA.

For more information about installing the Cloud Utility AWS v2.1.3 tool on the ESA, refer to the section *Working with the AWS Cloud Utility* in the [Protegity Appliances Overview Guide 9.1.0.5](#).

- The user accessing the **Cloud Utility AWS Tool** must have the **AWS Admin** role assigned on the ESA LDAP.

For more information about the AWS Admin role, refer to the section *Managing Roles* in the [Protegility Appliances Overview Guide 9.1.0.5](#).

► To install the Cloud Utility AWS tool:

1. Login to the DSG CLI Manager.
2. Navigate to **Administration > Add/Remove Services**.
3. Enter the root password.
4. Select **Install applications** and press **OK**.

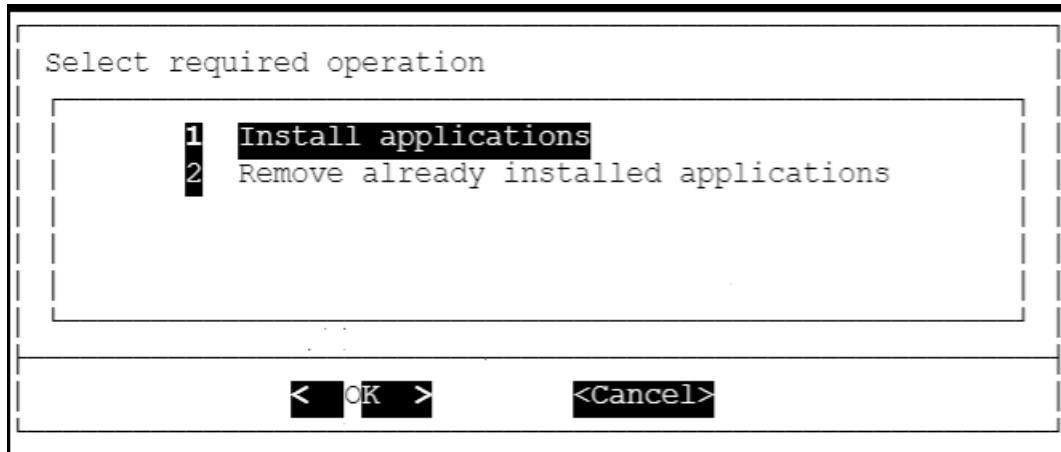


Figure 8-47: Installing application screen

5. Select **Cloud Utility AWS v2.1.3** and press **OK**.

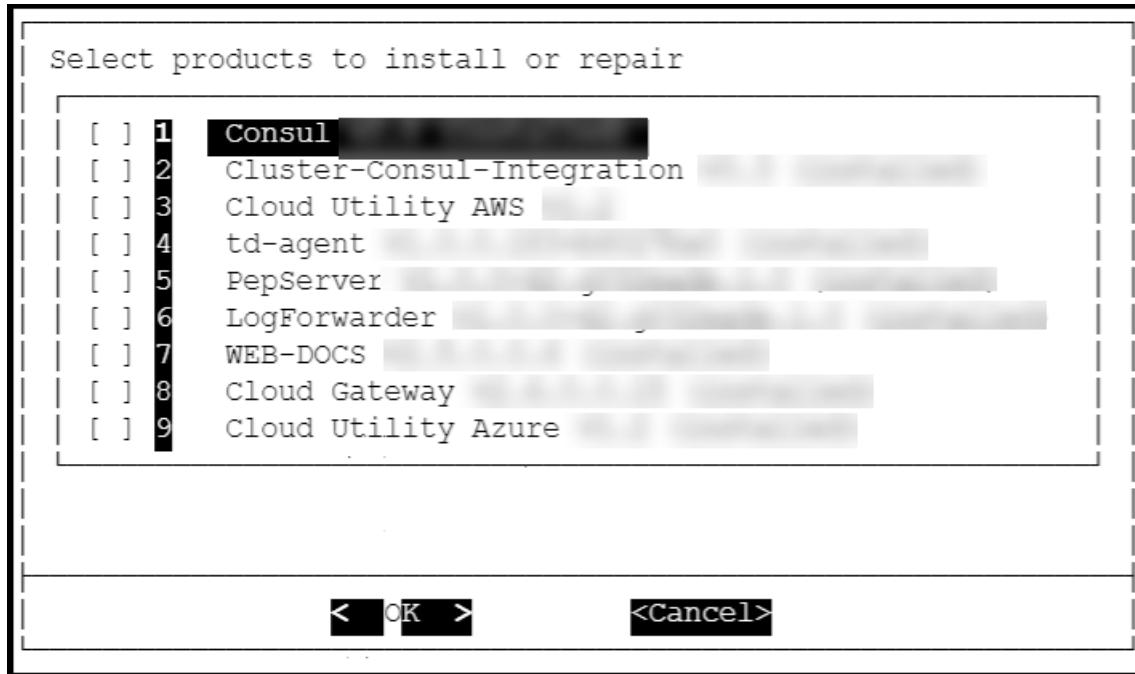


Figure 8-48: Adding the Cloud Utility AWS component

The Cloud Utility AWS v2.1.3 Tool component is added to the DSG instance.

8.2.5 Post DSG installation Steps

This section describes the steps to be performed after the finalization of the DSG installation is completed.

8.2.5.1 Verifying the DSG installation

This section describes the steps to verify the version details of the DSG instance.

► To verify the version details of the DSG instance:

1. Login to the DSG Web UI.
2. Click the  (Information) icon, and then click **About**.
3. Verify that the DSG version is reflected as *DSG 3.1.0.5*.

The DSG version details appear on the DSG Web UI successfully.

8.2.5.2 Pushing the DSG Rulesets

This section describes the steps to push the DSG Rulesets in a cluster.

► To push the DSG Rulesets:

1. Login to the ESA Web UI using the administrator credentials.
2. Navigate to **Cloud Gateway > Cluster**.
3. Go to **Actions** and click **Deploy** or **Deploy to Node Groups** to push the DSG Ruleset configurations to the DSG nodes.

Note: For more information about deploying the configurations, refer to the sections *Deploying the Configurations to Entire Cluster* or *Deploying the Configurations to Node Groups*.

The DSG Rulesets are pushed to the DSG nodes in a cluster or node groups.

8.2.5.3 Verifying the Startup Logs

This section describes the steps to verify the startup logs.

► To verify the DSG startup logs:

1. Login to the DSG Web UI using the administrator credentials.
2. Navigate to **Logs > Appliance**.
3. Click **Cloud Gateway - Event Logs**, and select **gateway**.
Verify that the startup logs do not display any errors.

For more information about handling error at startup, refer to the section *Appendix F: Troubleshooting Data Security Gateway (DSG)*.

The DSG startup logs are displayed on the DSG Web UI.

8.3 Extending ESA with DSG Web UI

This section describes the steps to install the *ESA_PAP-ALL-64_x86-64_9.1.0.5.xxxx.DSGUP.pty* patch on the ESA to extend its Web UI with the DSG menu.

► To install the DSG patch on the ESA Web UI:

1. Login to the ESA Web UI.
2. Navigate to **Settings > System > File Upload**.
3. Click **Choose File** to upload the DSG patch file.
4. Select the file and click **Upload**.
The uploaded patch appears on the Web UI.
5. On the ESA CLI Manager, navigate to **Administration > Installation and Patches > Patch Management**.
6. Enter the *root* password.
7. Select **Install a Patch** and press **OK**.
8. Select the uploaded patch.
9. Press **Install**.
The patch is successfully installed.

The DSG component is installed on the ESA. To verify the details of the DSG component from the *About* screen on the ESA, complete the following steps.

1. Login to the ESA Web UI.
2. Click the  (Information) icon, and then click **About**.
3. Verify that the DSG version is reflected as *DSG 3.1.0.5*.

8.4 Setting up ESA Communication

This section provides information to set ESA communication in cases, such as, change of the ESA IP address, change of the ESA certificates, adding the ESA IP address for cloud platforms, and joining the DSG in an existing cluster in the ESA.

► To setup the ESA communication:

1. On the DSG CLI Manager, navigate to **Tools > ESA Communication**.
2. Enter the root password, press **Tab** to select **OK** and press **Enter**.
3. Select all the options except **Join Cloud Gateway Cluster**, press **Tab** to select **Set Location Now** and press **ENTER**.

Caution:

Ensure that the **Join Cloud Gateway Cluster** option is deselected. It is recommended that the user add the DSG node from the **Cluster** tab after the DSG installation is completed.

For more information about adding a node to cluster, refer to the section [Adding a DSG node](#).

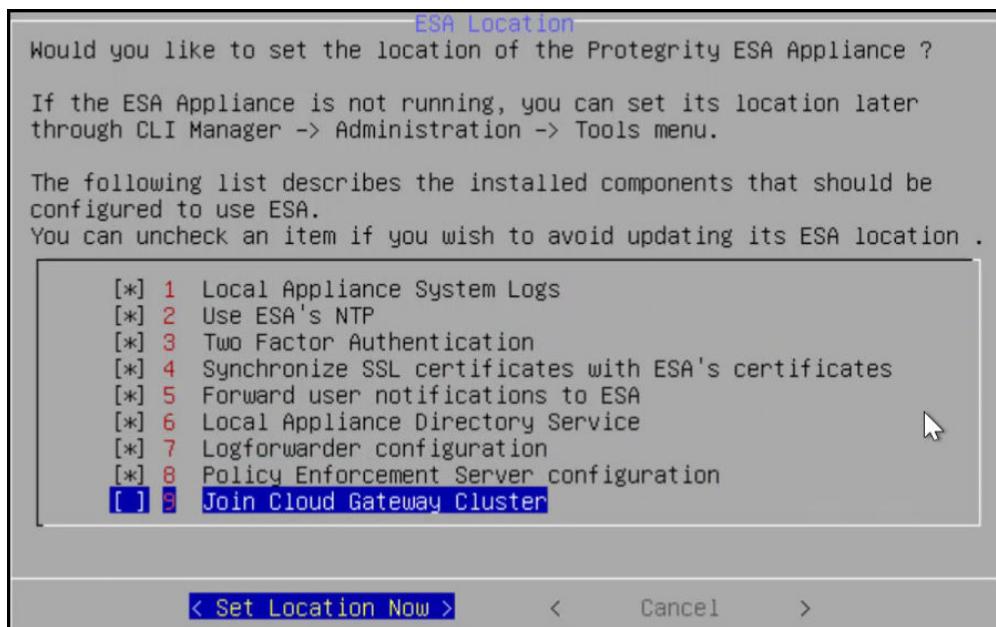


Figure 8-49: ESA Location screen

4. Select the ESA IP. Press **Tab** to select **OK** and press **Enter**.

The *ESA selection* screen appears.

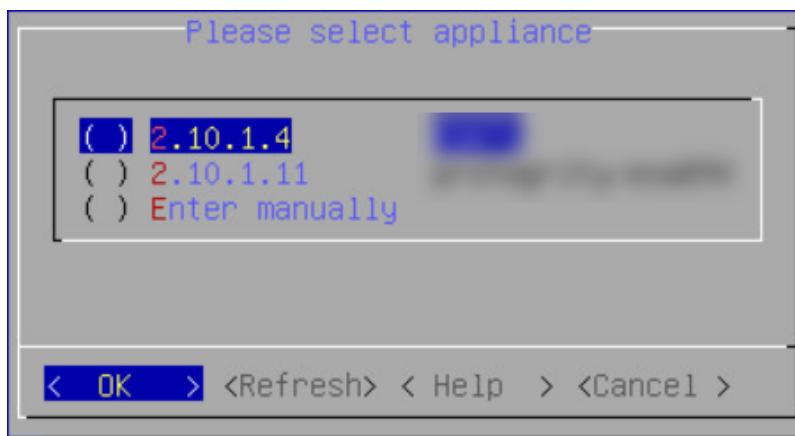


Figure 8-50: ESA appliance selection screen

Note: If you want to enter the ESA details manually, then select the **Enter manually** option. You will be asked to enter the ESA IP address or hostname when this option is selected.

5. Enter the ESA administrator credentials in the **Username** and **Password** text boxes. Press **Tab** to select **OK** and press **Enter**. The *Enterprise Security Administrator - Admin Credentials* screen appears.

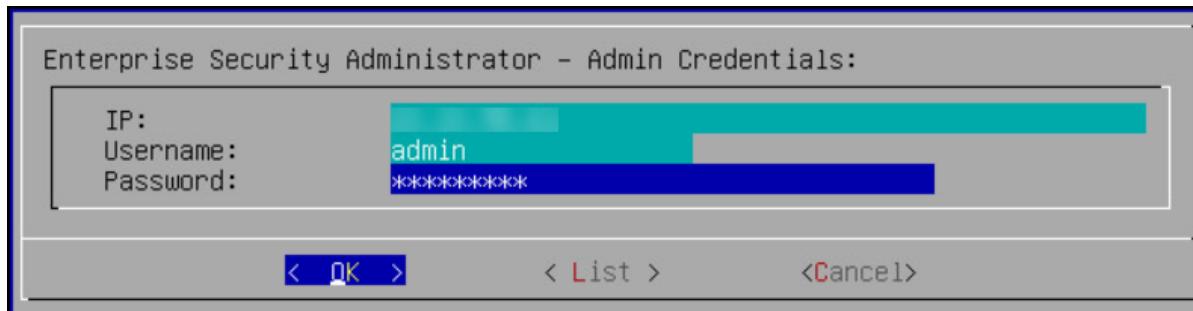


Figure 8-51: Enterprise Security Administrator - Admin Credentials screen

- Enter the IP address or hostname for the ESA. Press **Tab** to select **OK** and press **ENTER**. You can specify multiple IP addresses separated by comma.

The *Forward Logs to Audit Store* screen appears.

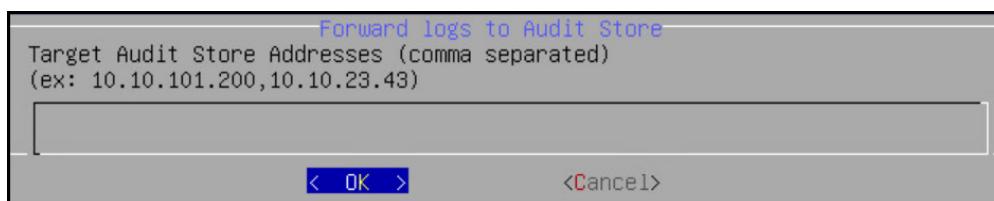


Figure 8-52: Forward Logs to Audit Screen

- After successfully establishing the connection with the ESA, the following Summary dialog box appears. Press **Tab** to select **OK** and press **Enter**.

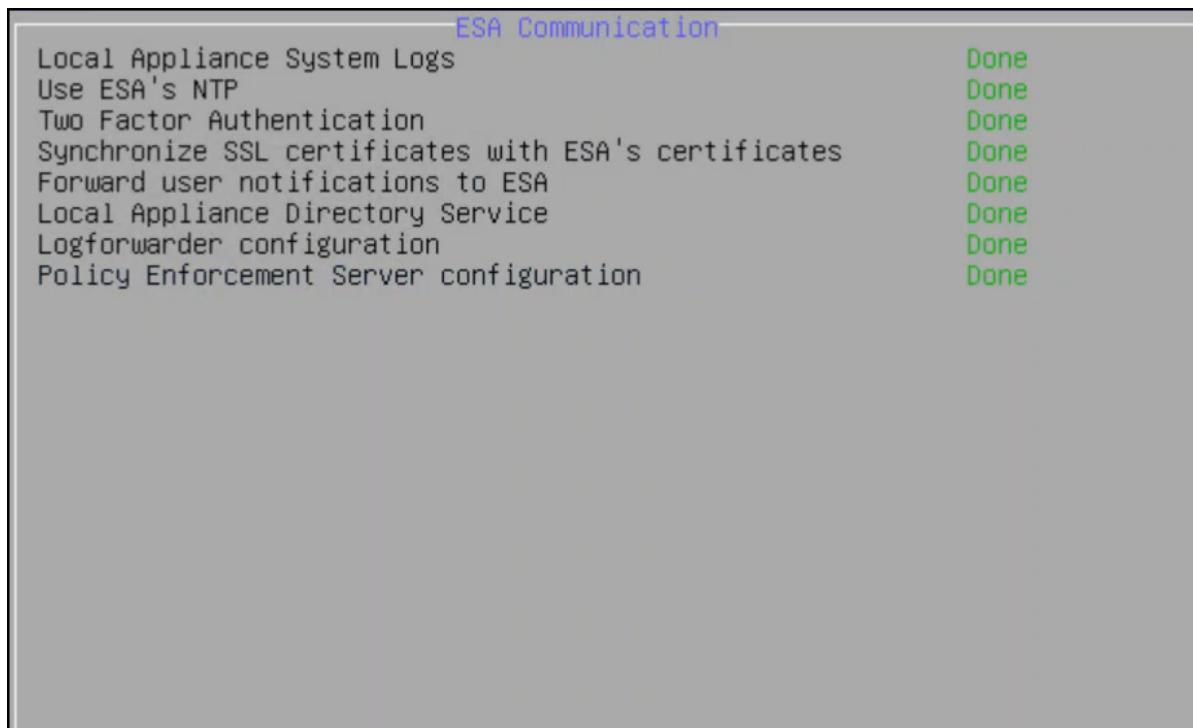


Figure 8-53: ESA Communication - Summary screen

The ESA communication is established successfully between the DSG and the ESA.

8.5 Configuring Default Gateway for Network Interfaces

When you install the DSG, to route network traffic, you can add an individual default gateway to the network interfaces in the same network using either the Static IP mode or the DHCP server. Mapping unique default gateways ensures requests intended for the required IP are routed through the default gateway assigned to that IP address.

8.5.1 Configuring Default Gateway for Management NIC (*ethMNG*) using the DSG CLI Manager

You can configure the default gateway for the management NIC (*ethMNG*) using the DSG CLI manager with the information provided in this section.

► To configure a default gateway for the Management NIC using the DSG CLI Manager:

1. On the DSG CLI Manager, navigate to **Networking > Network Settings**.
2. On the **Network Configuration Information** screen, press **Tab** to select **Interfaces** and press **Enter**.
3. Press **Tab** to select **ethMNG** and press **Enter** to add a default gateway.
4. Press **Tab** to select **Gateway** and press **Enter**.
5. Enter the Gateway IP address.
6. Press **Tab** to select **Apply** and press **Enter**.

8.5.2 Configuring Default Gateway for Service NIC (*ethSRV0*) using the DSG CLI Manager

You can configure the default gateway for the service NIC (*ethSRV0*) using the DSG CLI manager with the information provided in this section.

► To configure a default gateway to the Service NIC using the DSG CLI Manager:

1. On the DSG CLI Manager, navigate to **Networking > Network Settings**.
2. In the **Network Configuration Information** screen, press **Tab** to select **Interfaces** and press **Enter**.
3. Press **Tab** to select **ethSRV0** and press **Enter** to add a default gateway.
4. Press **Tab** to select **Gateway** and press **Enter**.
5. Enter the Gateway IP address.
6. Press **Tab** to select **Apply** and press **Enter**.

8.6 Configuring the DSG Cluster

To control and manage the DSG nodes from the ESA, create a cluster using the ESA Web UI and add the installed DSG nodes to it.

► To Create a Cluster:

1. On the ESA Web UI, navigate to **System > Trusted Appliances Cluster**.

The *Join Cluster* screen appears.

The screenshot shows the 'Join Cluster' interface. At the top, a descriptive text box says: 'Provide the IP of the target node along with credentials of the user with administrative privileges to connect with the target node. Then select a site and a preferred method to join a cluster.' Below this, there are three input fields: 'Target node IP', 'Username', and 'Password'. To the right of these fields are two buttons: 'Connect' and 'Clear'. At the bottom left is a button labeled 'Create a new Cluster', which is highlighted with a red rectangular border.

Figure 8-54: Join Cluster screen

2. Click **Create a new Cluster** to create a cluster.

The *Create Cluster* screen appears.

Create Cluster

A Trusted Appliances Cluster can be used to transfer data from one node to other nodes. Setting up a trusted appliances cluster allows you to synchronize files and configurations across multiple sites.

Communication Methods [i](#)

2.10.1.5

protegility-esa946.protegility.com

[Add New](#)

[Save](#)

[Join a Cluster](#)

Figure 8-55: Create Cluster screen

3. Select a preferred communication method.

Note: For more information about the communication methods, refer to the section *Managing Local to Remote Node Communication* in the *Protegility Appliances Overview Guide 9.1.0.5*.

4. Click **Save** to create a cluster.

Note: The ESA administrator username and password are used automatically when you click **Save**. The cluster is created.

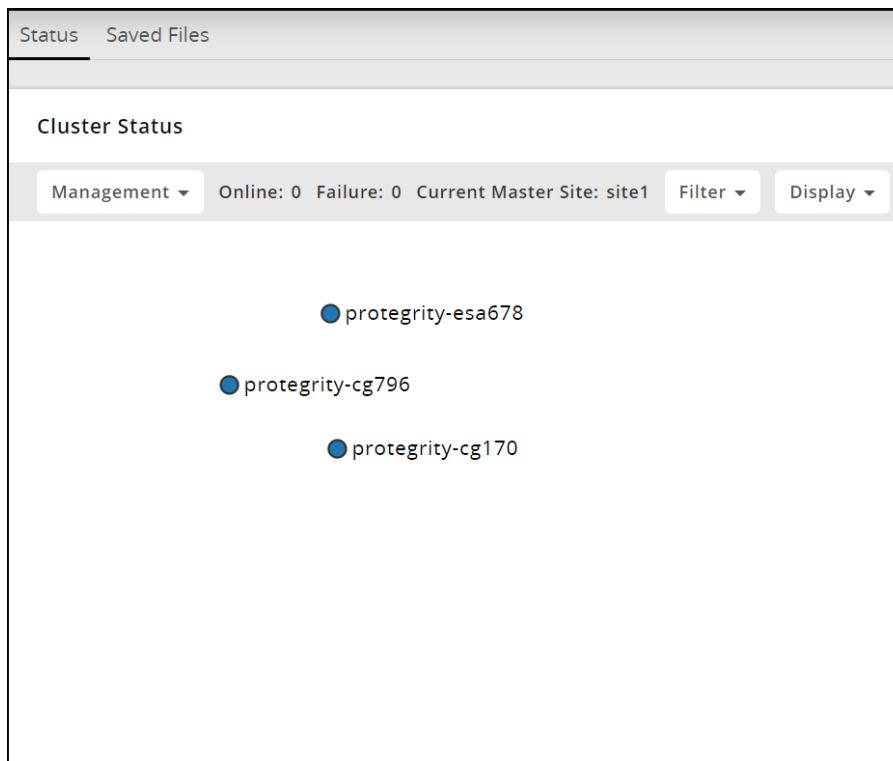


Figure 8-56: Cluster screen

8.6.1 Adding a Node to the Cluster

The *Cluster Monitoring* screen lets you add nodes to a cluster such that the *Cluster Monitoring* screen shows the cluster status for each node at a glance.

Note: Ensure that the Trusted Appliance Cluster (TAC) is created on ESA and then you add the DSG node to the cluster.

For more information about creating a TAC, refer to the section [Create a TAC](#).

► To add a Node to the Cluster:

1. On the ESA Web UI, navigate to **Cloud Gateway > Cluster > Monitoring**.

The *Cluster* screen appears. No nodes are added to the cluster.



Figure 8-57: Cluster Monitoring screen

2. Select the **Actions** drop down list in the Cluster Health pane.

The following options appear:

- Apply Patch on Cluster

- Apply Patch on selected Nodes
 - Change Groups on Entire Cluster
 - Change Groups on Selected Nodes
 - Add Node
3. Perform the following steps to add a node.
- a. Click **Add Node**.

The *Add new node to cluster* screen appears.

The screenshot shows a modal dialog titled "Add new node to cluster". It contains the following fields:

- Node IP***: An input field for the node's IP address.
- Node User Name***: An input field for the administrator user name.
- Node Password***: An input field for the administrator password.
- Deployment Node Group i**: A dropdown menu showing "default".
- Sites :** A dropdown menu showing "site1".

At the bottom right are two buttons: "Submit" (dark grey) and "Close" (light grey).

Figure 8-58: Add New Node screen

- b. Enter the DSG node IP address to be added in the cluster in the **Node IP** field.
- c. Enter the administrator user name for the ESA node user in the **Node User Name** field.
- d. Enter the administrator password for the ESA node user in the **Node Password** field.
- e. Enter the node group name in the **Deployment Node Group** field.

Note: If the deployment node group is not specified, by default it will get assigned to the *default* node group.

- f. Click **Submit**.
- g. Click **Refresh > Deploy or Deploy to Node Groups**.

Note: For more information about deploying the configurations to entire cluster or the node groups, refer to the section [Deploying the Configurations to Entire Cluster](#) and [Deploying the Configurations to Node Groups](#).

The node is added to the cluster.

The screenshot shows a table titled "Healthy Nodes" with one entry. The columns are: Hostname, IP, PAP Version, Health, Node Group, Config Version, DSG Version, Uptime, and Actions. The node listed is "protegility-cg170" with IP "10.36.1.202", PAP Version "9.0.0.0", Health status "green", Node Group "default", Config Version "3.0.0.0", DSG Version "3.0.0.0", and Uptime "1 day ago". There is a "Select for Patch Install or Node Group Update" button at the top right.

Figure 8-59: Cluster Monitoring screen

The following figure is the Trusted Appliances Cluster (TAC) page after adding the nodes to the cluster.

The screenshot shows the "Cluster Status" section of the TAC page. It lists three nodes: "protegility-esa678", "protegility-cg170", and "protegility-cg796". To the right, there is a detailed table for each node with the following information:

Name	protegility-cg170(10.36.1.202)
Description	Created on Wed Nov 24 02:40:36 2021
Labels	_all_,CG_Cloud_Gateway,site1,dsq,dsq_default
Status	Online
Status Message	No messages

Name	protegility-cg796(10.36.1.179)
Description	Created on Mon Nov 22 14:20:25 2021
Labels	_all_,CG_Cloud_Gateway,site1,dsq,job1,Consul Client
Status	Online
Status Message	No messages

Name	protegility-esa678(10.36.1.178)
Description	Created on Mon Nov 22 14:08:36 2021
Labels	_all_,ESA_Enterprise-Security-Administrator,site1,Consul Server
Status	Online
Status Message	No messages

Figure 8-60: TAC with the DSG nodes

8.6.2 Applying a DSG Patch

This section describes the steps to apply a DSG patch on the ESA and the DSG nodes.

Before you begin

Note: Ensure that any patch application is performed using the Integrated Lights-Out (iLO) interface/VM console. Do not install this patch using an SSH connection.

Attention: You must install this patch on the individual ESA or DSG nodes only from the CLI Manager. You cannot apply a patch to multiple DSG nodes from the *Cluster Monitoring* screen in ESA.

► To apply the DSG patch:

1. Login to the ESA/DSG Web UI.
2. Navigate to **Settings > System > File Upload**.
3. Click **Choose File** to upload the patch file.
4. Select the file and click **Upload**.
The uploaded patch appears on the ESA/DSG Web UI.
5. On the ESA/ DSG CLI Manager, navigate to **Administration > Installation and Patches > Patch Management**.
6. Enter the *root* password.
7. Select **Install a Patch** and press **OK**.

8. Select the uploaded patch and press **OK**.
 9. Press **Install**.
- The DSG patch is installed.

8.6.3 Applying an Appliance Framework Patch

This section describes the steps to apply an appliance framework patch.

Before you begin

Note: Ensure that any patch application is performed using the Integrated Lights-Out (iLO) interface/VM console. Do not install this patch using an SSH connection.

► To apply the appliance framework patch:

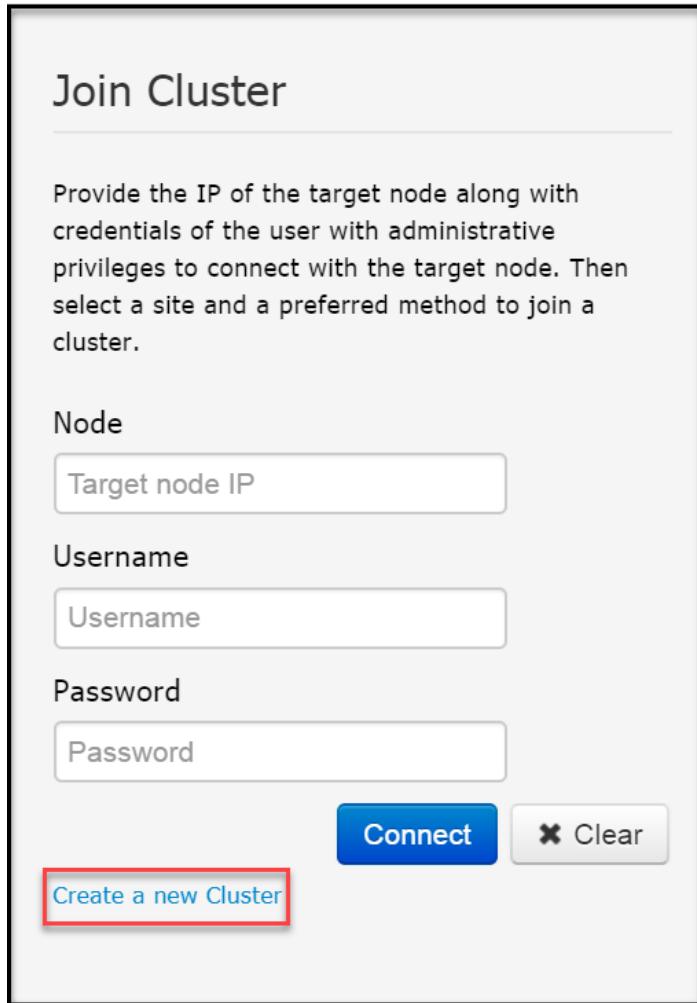
1. Login to the DSG Web UI.
2. Navigate to **Settings > System > File Upload**.
3. Click **Choose File** to upload the patch file.
4. Select the file and click **Upload**.
The uploaded patch appears on the DSG Web UI.
5. On the DSG CLI Manager, navigate **Administration > Installation and Patches > Patch Management**.
6. Enter the *root* password.
7. Select **Install a Patch** and press **OK**.
8. Select the uploaded patch and press **OK**.
9. Press **Install**.
The patch is applied to the DSG node.

8.6.4 Creating a Cluster

To control and manage the DSG nodes from the ESA, create a cluster using the ESA Web UI and add the installed DSG nodes to it.

► To Create a Cluster:

1. From the ESA, navigate to **System > Trusted Appliances Cluster**.
The *Join Cluster* screen appears.



The image shows the 'Join Cluster' screen. At the top, the title 'Join Cluster' is displayed. Below it, a descriptive text instructs the user to provide the IP of the target node and administrative credentials to connect with the target node, and to select a site and a preferred method to join a cluster. The form is divided into sections for 'Node', 'Username', and 'Password'. Each section contains a text input field. Below these fields are two buttons: a blue 'Connect' button and a grey 'Clear' button with a clear icon. At the bottom left, there is a link labeled 'Create a new Cluster' which is highlighted with a red rectangular border.

Figure 8-61: Join Cluster screen

2. Click the **Create a new Cluster** link.
3. Click **Create** to create a cluster.

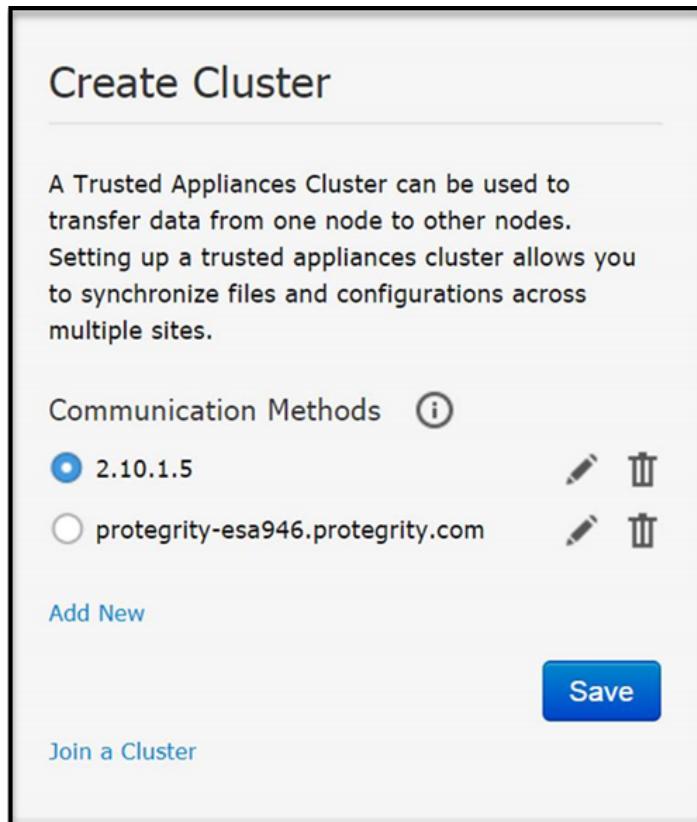


Figure 8-62: Create Cluster screen

The *Create Cluster* screen appears.

4. Select a preferred communication method.

Note: For more information about communication methods, refer the section *Managing Communication Methods for Local Node* in the [Protegility Appliances Overview Guide 9.1.0.5](#).

5. Click **Save** to create a cluster.

Note: The ESA administrator username and password is used automatically when you click **Save**. The cluster is created.

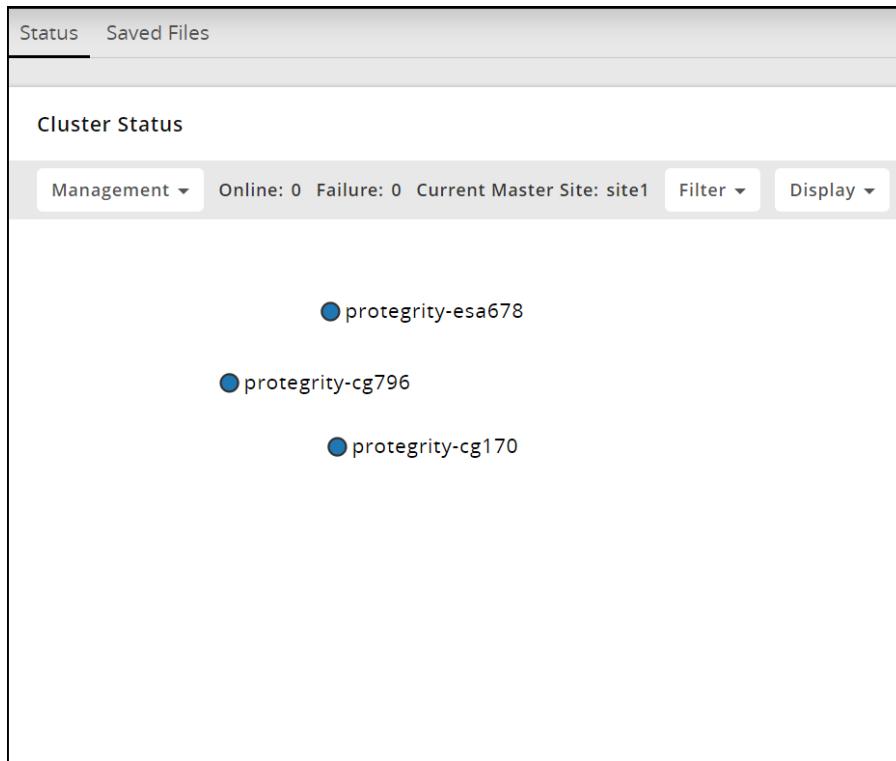


Figure 8-63: Cluster screen

8.6.5 Removing a Node from the Cluster

If you no longer intend to use a node added to the cluster, then you can use the steps in this section.

► To remove a Node from the Cluster:

1. On the ESA Web UI, navigate to **Cloud Gateway > Cluster > Monitoring**.
2. Click **Actions** next to the node you want to remove.
3. Click **Delete**.

The node is removed from the cluster.

8.7 Forwarding Logs to the Audit Store

After installing or upgrading to the DSG 3.1.0.4, you must configure the DSG to forward the DSG logs to the Audit Store on the ESA using the steps provided in this section.

Before you begin

Ensure that you have configured the *Audit Store* component on the ESA. Configuring this component allows Forensics to retrieve the DSG appliance and audit logs.

For more information about *Audit Store*, refer to the section *Understanding the Audit Store* in the *Protegility Log Forwarding Guide 9.1.0.5*.

8.7.1 Forwarding Appliance Logs to the Audit Store

The appliance logs are the syslog, which are forwarded through the *td-agent* service to the Audit Store on the ESA.

► To forward appliance logs to the Audit Store:

1. Login to the DSG CLI Manager.
2. Navigate to **Tools > PLUG - Forward logs to Audit Store**.

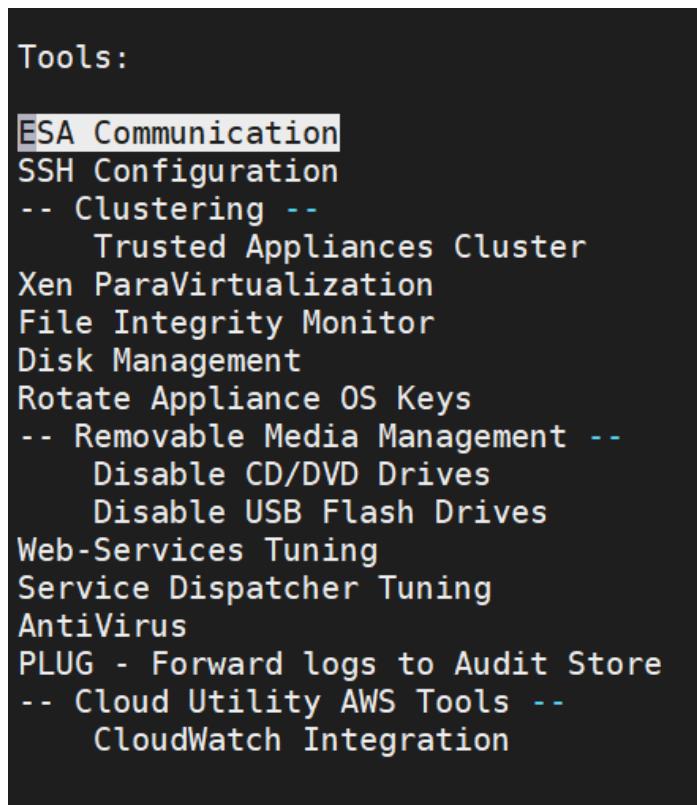


Figure 8-64: Forwarding Logs to Audit Store

3. Enter the password of the DSG root user and select **OK**.

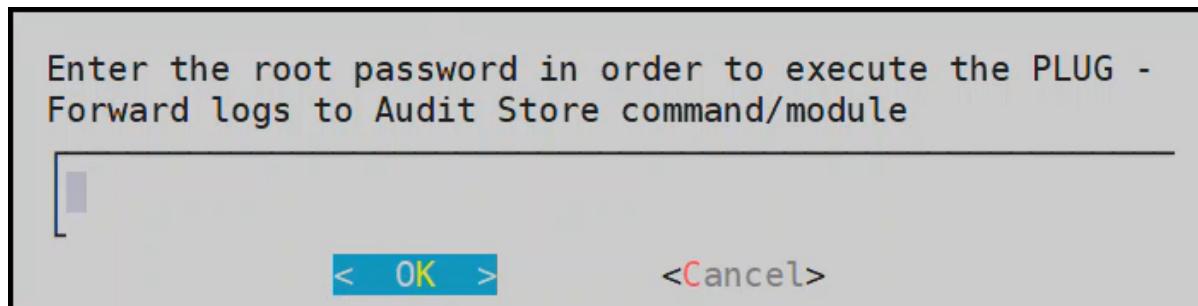


Figure 8-65: Root Password Screen

4. Enter the username and password of the DSG administrator user and select **OK**.

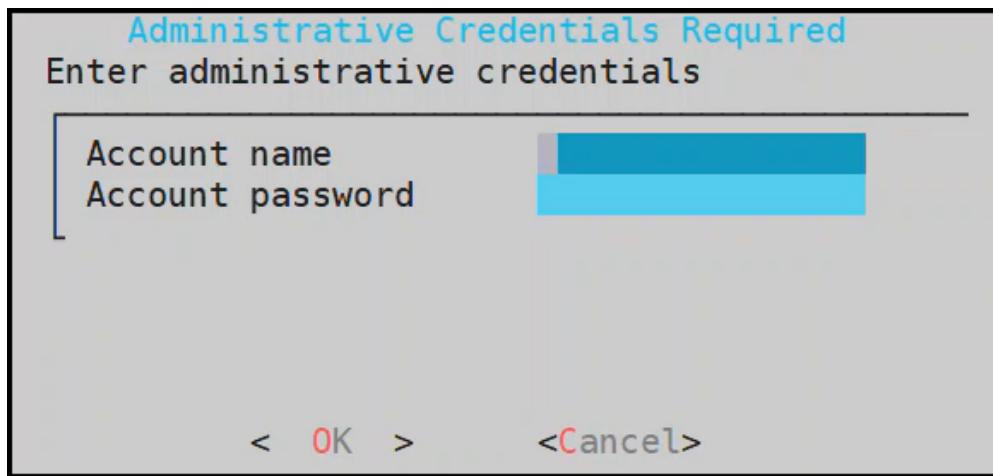


Figure 8-66: ESA Administrator Credentials screen

5. Select **OK**.

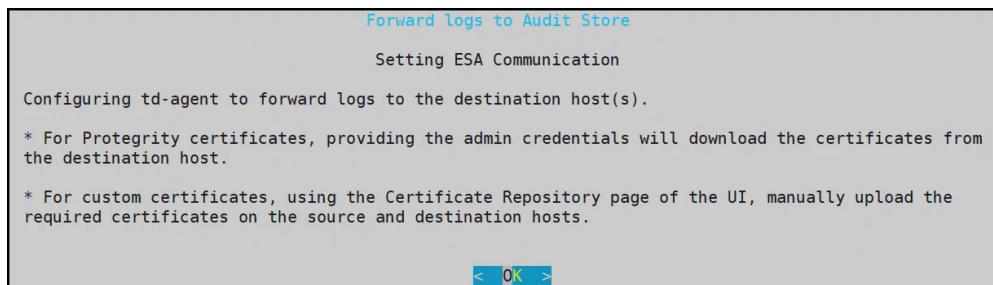


Figure 8-67: Setting ESA Communication

6. Enter the IP address for the ESA and select **OK**. You can specify multiple IP addresses separated by comma.

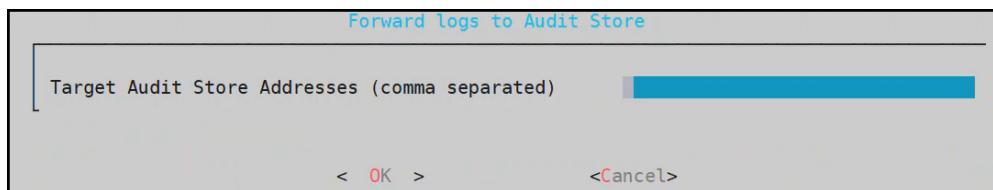


Figure 8-68: Forward Logs to Audit Screen

7. Enter y to fetch certificates and select **OK**.

These certificates is used to validate and connect to the target node. It is required to authenticate with the Audit Store while forwarding logs to the target node.

If the certificates already exists on the system, then specify n in this screen.

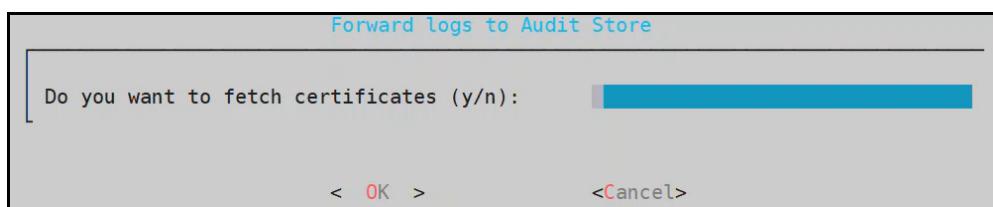


Figure 8-69: Audit Store Certificate

8. Enter the username and password of the ESA administrator user and select **OK**.

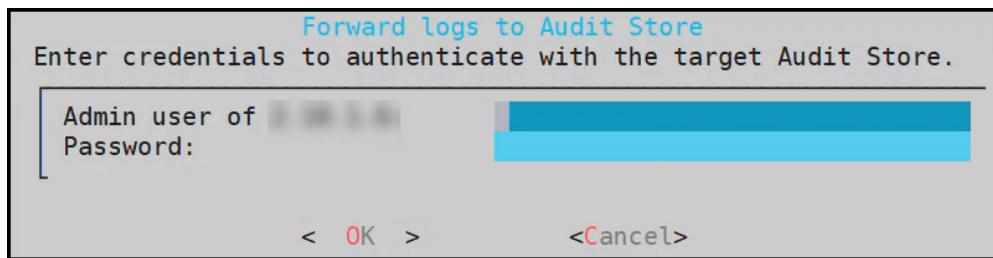


Figure 8-70: Audit Store Administrator User Password screen

The *td-agent* service is configured to send logs to the Audit Store and the CLI menu appears.

9. Repeat step 1 to step 8 on all the DSG nodes in the cluster.

8.7.2 Forwarding Audit Logs to the Audit Store

The audit logs are the data security operation-related logs, namely protect, unprotect, and reprotect and the PEP server logs. The audit logs are forwarded through the *Log Forwarder* service to the Audit Store on the ESA.

► To forward audit logs to the Audit Store:

1. Login to the DSG CLI Manager.
2. Navigate to **Tools > ESA Communication**.

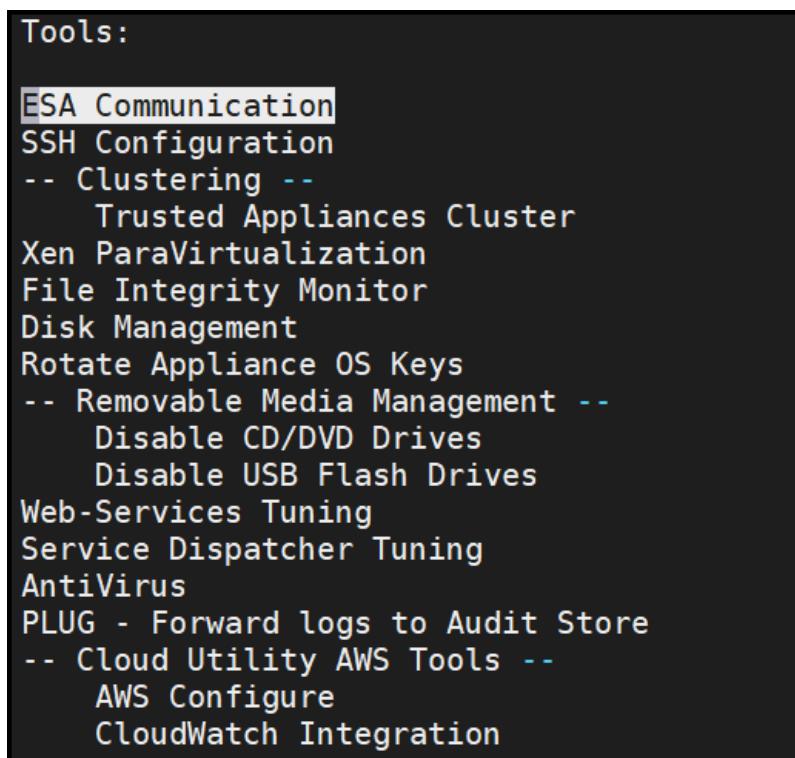


Figure 8-71: Forwarding Audit Logs to Audit Store

3. Enter the password of the DSG root user and select **OK**.

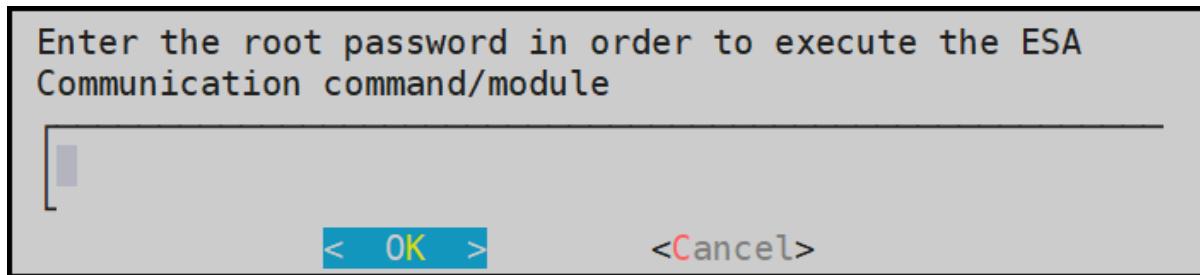


Figure 8-72: Root Password Screen

4. Select the *Logforwarder configuration* option. Press **Tab** to select **Set Location Now** and press **Enter**.
The *ESA Location* screen appears.

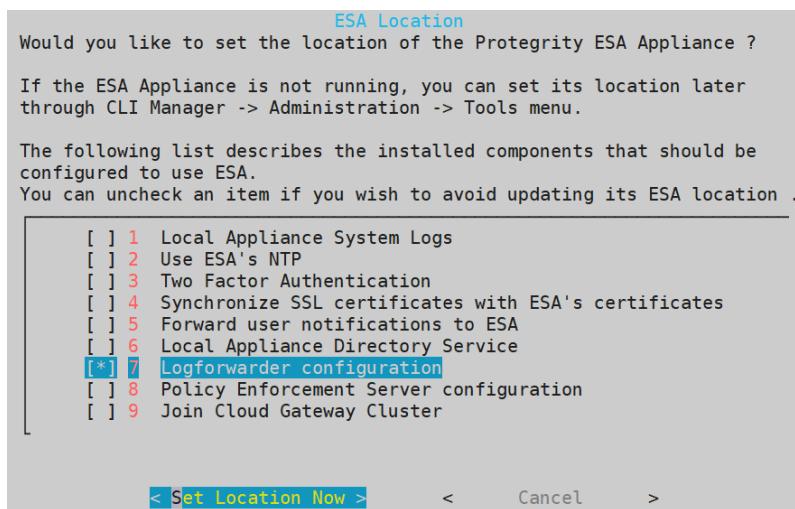


Figure 8-73: Setting the ESA location

5. Select the ESA that you want to connect with, and then press **Tab** to select **OK** and press **ENTER**.
The *ESA selection* screen appears.

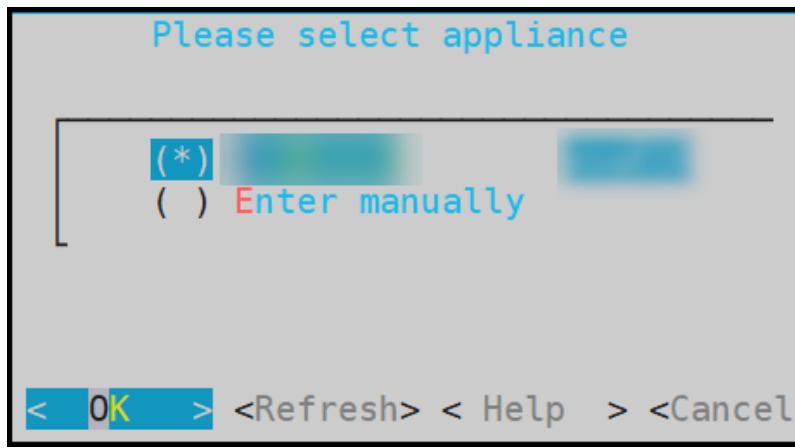


Figure 8-74: ESA appliance selection screen

Note: If you want to enter the ESA details manually, then select the **Enter manually** option. You will be asked to enter the ESA IP address or hostname when this option is selected.

6. Enter the ESA administrator username and password to establish communication between the ESA and the DSG. Press **Tab** to select **OK** and press **Enter**.

The *Enterprise Security Administrator - Admin Credentials* screen appears.

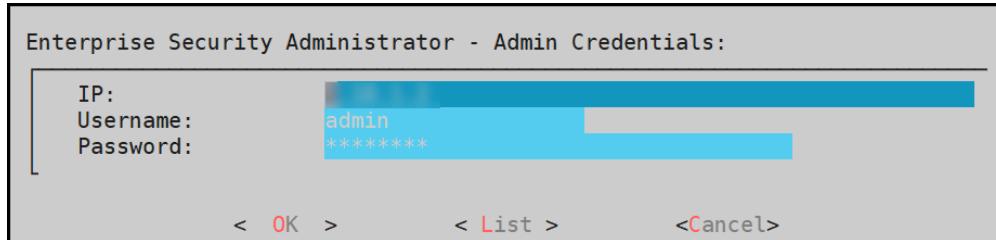


Figure 8-75: Enterprise Security Administrator - Admin Credentials screen

- Enter the IP address or hostname for the ESA. Press **Tab** to select **OK** and press **ENTER**. You can specify multiple IP addresses separated by comma.

The *Forward Logs to Audit Screen* appears.

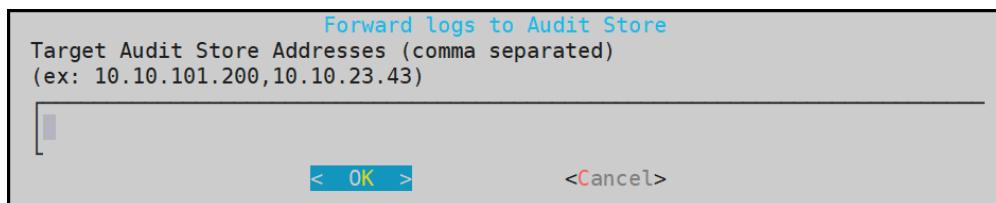


Figure 8-76: Forward Logs to Audit Screen

- After successfully establishing the connection with the ESA, the following Summary dialog box appears. Press **Tab** to select **OK** and press **Enter**.



Figure 8-77: ESA Communication - Summary screen

- Repeat step 1 to step 8 on all the DSG nodes in the cluster.

8.8 Advanced Settings

After you install and setup the DSG cluster, you can configure the advanced settings listed in this section to improve the performance of your setup.

8.8.1 Forking DSG Processes

The performance of the DSG setup is based on the number of CPU cores. Typically, if the number of CPU cores on an appliance is 16, then the DSG forks 16 processes as a default. You can fork the DSG processes to a lesser number than the default number, if you intend to utilize system resources effectively.

► To limit the number of DSG processes on a single DSG appliance:

1. From the ESA CLI Manager, navigate to **Administration > OS Console**.
2. Navigate to the `/etc/init.d` directory.
3. Open the `allianceGateway` file for editing.
4. Edit the following parameter to add the number of processes to fork.

```
-- python -m ${ALLIANCE_DIR}/bin/gateway $1 --configDir $CONFIG_DIR --fork N
```

In this command, *N* is the number of processes that you want to fork.

Note: By default, the DSG appliance forks the number of processes equivalent to the number of CPU cores on the appliance.

5. Restart the DSG appliance for the updates to take effect.

Chapter 9

Upgrading to DSG v3.1.0.5

- [9.1 Backing Up the DSG Appliance OS from the Web UI](#)
- [9.2 Upgrading to DSG v3.1.0.5 from DSG v3.1.0.4](#)
- [9.3 Upgrading to DSG v3.1.0.5 from DSG v3.1.0.3](#)
- [9.4 Upgrading to DSG v3.1.0.5 from DSG v3.1.0.2](#)
- [9.5 Upgrading to DSG v3.1.0.5 from DSG v3.1.0.1](#)
- [9.6 Upgrading to DSG v3.1.0.5 from DSG v3.1.0.0](#)
- [9.7 Upgrading to DSG v3.1.0.5 from DSG v3.0.0.0](#)
- [9.8 Upgrading to DSG v3.1.0.5 from DSG v2.6.0.1](#)
- [9.9 Upgrading to DSG v3.1.0.5 from DSG v2.6.0.0](#)
- [9.10 Upgrading to DSG v3.1.0.5 from DSG v2.4.2](#)
- [9.11 Upgrading to DSG v3.1.0.5 from DSG v2.4.1](#)
- [9.12 Upgrading to DSG v3.1.0.5 from DSG v2.4.0](#)
- [9.13 Restoring the DSG Appliance OS Backup](#)
- [9.14 Pushing Audit Logs to Forensics](#)
- [9.15 Managing the PEP Server Configuration File](#)
- [9.16 Restarting the DSG Node](#)
- [9.17 Restoring LogForwarder Custom Files](#)
- [9.18 Restoring the Backed Up Codebook Reshuffling Configuration files](#)
- [9.19 Verifying UDF Rules for Blocked Modules and Methods](#)
- [9.20 Products Compatibility Matrix](#)

This section provides the order of installation to upgrade to the Data Security Gateway (DSG) v3.1.0.5 from various DSG versions that are installed on-premise and on cloud platforms.

Note:

Ensure that the ESA is upgraded to the ESA v9.1.0.5. before you upgrade the DSG version to the DSG v3.1.0.5.

For more information about upgrading the ESA, refer to the section *Upgrade Paths to ESA v9.1.0.5* in the *Protegility Upgrade Guide 9.1.0.5*.

The following figure provides the upgrade path that you must follow to upgrade the DSG version to the DSG v3.1.0.5 on the ESA.

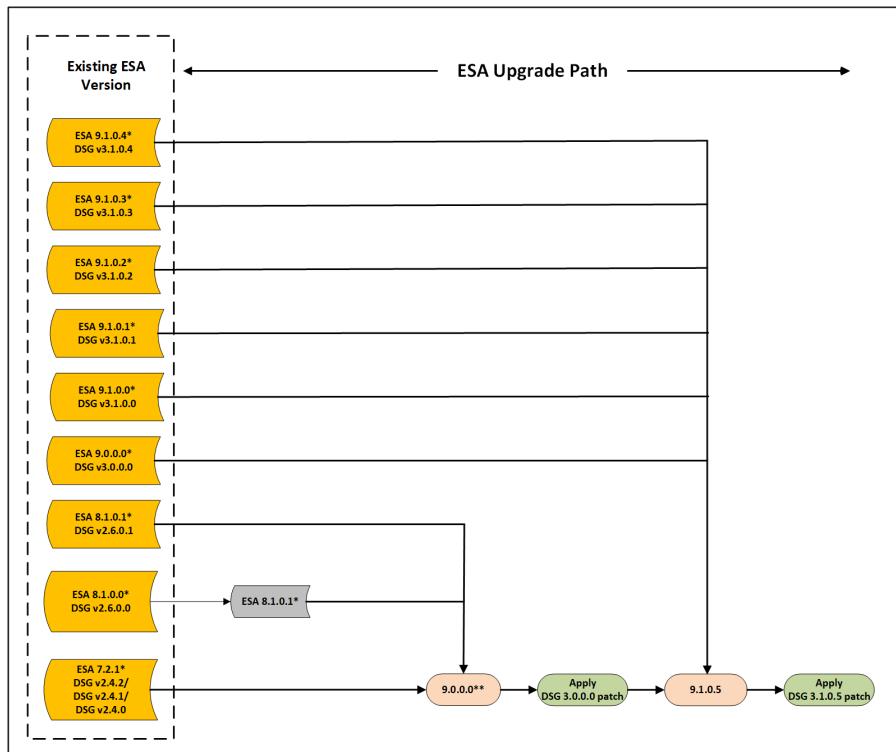


Figure 9-1: Upgrade Path - DSG Component on ESA

Caution: * indicates all the available hotfix and security patches on the platform version.

** indicates the *pre-v9.0.0.0* patch. This is an intermediate step while migrating from v7.2.1 to v9.0.0.0. This *pre-v9.0.0.0* patch should not be used for upgrading to any other version.

Note: For more information about upgrading the ESA to v9.1.0.5, refer to the section *Upgrade Paths to ESA v9.1.0.5* in the *Protegility Upgrade Guide 9.1.0.5*.

Note:

Ensure that the corresponding ESA hotfix patch is installed on the ESA.

The following figure provides the upgrade path that you must follow to upgrade the older versions of DSG to DSG v3.1.0.5 on a DSG node.

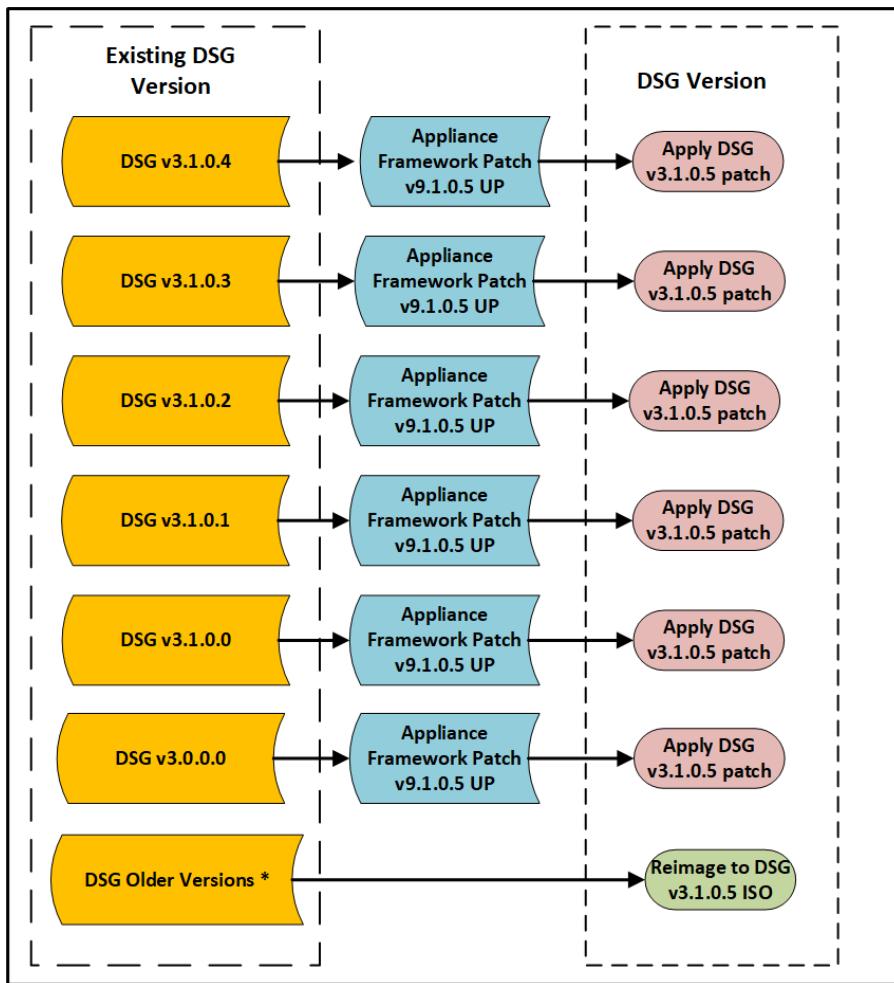


Figure 9-2: Upgrade Path - DSG Node

Note:

* indicates DSG v2.x.x releases

9.1 Backing Up the DSG Appliance OS from the Web UI

This section describes the steps to backup the DSG Appliance OS from the Web UI.

► To backup the DSG Appliance OS from the Web UI:

1. Login to the DSG Web UI.
2. Navigate to **System > Backup & Restore**.
3. Navigate to the **OS Full** tab and click **Backup**.
The message *The backup process may take several minutes to complete. It is recommended to stop all services prior to starting the backup* appears.
4. Press **ENTER**.
The *Backup Center* screen appears and the OS backup process is initiated.

5. Navigate to the DSG Dashboard.

The notification *O.S Backup has been initiated* appears. After the backup is complete, the notification *O.S Backup has been completed* appears.

9.2 Upgrading to DSG v3.1.0.5 from DSG v3.1.0.4

This section describes the steps to upgrade to the DSG v3.1.0.5 from DSG v3.1.0.4.

Note:

You can upgrade from any version of DSG v3.1.0.4, including Hotfixes (HFs), Feature Enhancements (FEs), and Security Enhancements (SEs) to DSG v3.1.0.5.

Ensure that an ESA v9.1.0.4 with the DSG v3.1.0.4 patch installed is available.

Note:

For more information about upgrading the DSG component on the ESA, refer to the section [Upgrading to DSG v3.1.0.4](#).

Ensure that the DSG v3.1.0.4 is installed.

Note:

For more information about installing DSG v3.1.0.4 appliance, refer to the section *Installing the DSG* in the [Protegility Data Security Guide 3.1.0.4](#).

Before you begin:

Ensure that the following prerequisites are met before you upgrade the DSG to v3.1.0.5:

- Verify if the `pepservera<x>.dat` and `pepserverb<x>.dat` file size in the `/opt/protegility/defiance_dps/data/` directory is *zero*. If the file size is greater than *zero*, then refer to the section [Backing Up the DSG Appliance OS from the Web UI](#) to ensure that all the audit logs are successfully sent to the Forensics before the DSG upgrade.

For more information about sending the audit logs to the Forensics, refer to the section [Pushing Audit Logs to Forensics](#).

- For each DSG node in the cluster, ensure that the `pepserver.cfg` configurations are saved before reimaging the DSG. These configurations must be manually set after the DSG is reimaged.

For more information about downloading the PEP server configuration file, refer to the section [Managing the PEP Server Configuration File](#).

- It is recommended that the Ruleset definitions are configured on the ESA such that the same configuration can be pushed simultaneously to all the nodes in the cluster.
- It is also recommended that any import or export of the DSG configurations and Ruleset configurations are performed from the ESA Web UI.

Caution:

The export and import of the DSG configurations and Ruleset configurations to DSG v3.1.0.5 is supported from the DSG v2.4.2 and higher. Navigate to **Settings > Backup & Restore** to export/import the DSG configurations and Ruleset configurations.

- Note:** Ensure that you do not import any DSG Ruleset backups (`.zip`) from the older DSG versions to the DSG 3.1.0.5.



- Caution:** If you are using codebook reshuffling, then ensure that for each DSG node where codebook reshuffling is configured, the following files are backed up before upgrading to the DSG 3.1.0.5:
 - *BLOB (random.dat)*
 - *dps.env*
 - *User PIN (userpin.bin)*

After backing up the files, you must create a *.tgz* package, which consists of all these files. Run the following command to create a *.tgz* package.

```
tar --same-owner -zcpvf /products/
uploads/<filename>.tgz /opt/protegility/defiance_dps/data/random.dat /opt/protegility/
defiance_dps/data/userpin.bin /opt/protegility/defiance_dps/bin/dps.env
```

Run the following command to set the required permissions for downloading the *.tgz* package from the DSG Web UI.

```
chmod 644 /products/uploads/<filename>.tgz
```

Perform the following steps to download the *.tgz* package from the DSG Web UI.

1. Login to the DSG Web UI.
2. On the DSG Web UI, navigate to **Settings > System > File Upload**.
3. Select the *.tgz* file from the *Uploaded Files* drop-down and click **Download**.

- Ensure that the *ESA_PAP-ALL-64_x86-64_9.1.0.5.xxxx.DSGUP.pty* patch is available. This patch is applied on the ESA to extend the ESA with the DSG Web UI.
- Ensure that the *DSG_PAP-ALL-64_x86-64_3.1.0.5.x.UP.tgz* file is available. This file contains the *DSG_PAP-ALL-64_x86-64_3.1.0.5.x.UP.pty* and *PAP_PAP-ALL-64_x86-64_9.1.0.5.x.UP.pty* patch files.

Caution:

Ensure that you do not uninstall any Cloud Gateway service from the ESA.

Upgrading to the DSG v3.1.0.5 on ESA Node

Ensure that you use the following upgrade order to install the DSG v3.1.0.5 patch.

Table 9-1: Order of Upgrade for the ESA

Order of upgrade	Description	Reference
1	Upgrade ESA 9.1.0.4 to ESA 9.1.0.5.	Section Upgrading to v9.1.0.x in the <i>Protegility Upgrade Guide 9.1.0.5</i> .
2	Apply the DSG v3.1.0.5 patch (<i>ESA_PAP-ALL-64_x86-64_9.1.0.5.xxxx.DSGUP.pty</i>) on the ESA v9.1.0.5.	<i>Applying the DSG 3.1.0.5 Patch</i>

Upgrading to the DSG v3.1.0.5 from the DSG v3.1.0.4 Node

Ensure that you use the following table to upgrade the DSG nodes to the DSG v3.1.0.4.

Table 9-2: Order of Upgrade for the DSG

Order of installation	Description	Reference
1	Apply the <i>PAP_PAP-ALL-64_x86-64_9.1.0.5.x.UP.pty</i> Appliance Framework patch.	<i>Applying the Appliance Framework patch</i>



Order of installation	Description	Reference
2	Apply the <i>DSG_PAP-ALL-64_x86-64_3.1.0.5.x.UPpty</i> patch on the DSG 3.1.0.4.	Applying the DSG 3.1.0.5 Patch

Caution: If you are using codebook reshuffling, then ensure that the following codebook reshuffling configuration files for each DSG node are restored to the respective DSG node after upgrading to DSG 3.1.0.5:

- *BLOB (random.dat)*
- *dps.env*
- *User PIN (userpin.bin)*

For more information about configuring the backed up codebook reshuffling configuration, refer to the section [Restoring the Backed Up Codebook Reshuffling Configuration files](#).

Note:

Ensure that the PEP server configurations on each DSG node are modified manually according to the *pepserver.cfg* file saved for the respective DSG node using the information in the section [Managing the PEP Server Configuration File](#).

Note:

If any changes are made on a DSG node in the cluster, then create a scheduler task to replicate policies, configuration, DSG rulesets, and so on, from the DSG having changes to the other DSGs in the cluster.

For more information about creating a cluster task, refer to the section *Scheduling Configuration Export to Cluster Tasks* in the [Protegility Appliances Overview Guide 9.1.0.5](#).

Note:

If you are using UDFs in Rule definitions, then verify whether any of the blocked modules and methods are a part of your definitions.

For more information about handling UDFs in Rule definitions with blocked modules and methods, refer to the section [Verifying UDF Rules for Blocked Modules and Methods](#).

9.3 Upgrading to DSG v3.1.0.5 from DSG v3.1.0.3

This section describes the steps to upgrade to the DSG v3.1.0.3 from DSG v3.1.0.3.

Note:

You can upgrade from any version of DSG v3.1.0.3, including Hotfixes (HFs), Feature Enhancements (FEs), and Security Enhancements (SEs) to DSG v3.1.0.5.

Ensure that an ESA v9.1.0.3 with the DSG v3.1.0.3 patch installed is available.

Note:

For more information about upgrading the DSG component on the ESA, refer to the section [Upgrading to DSG v3.1.0.3](#).

Ensure that the DSG v3.1.0.3 is installed.

Note:

For more information about installing DSG v3.1.0.3 appliance, refer to the section *Installing the DSG* in the *Protegility Data Security Guide 3.1.0.3*.

Before you begin:

Ensure that the following prerequisites are met before you upgrade the DSG to v3.1.0.5:

- Verify if the `pepservera<x>.dat` and `pepserverb<x>.dat` file size in the `/opt/protegility/defiance_dps/data/` directory is *zero*. If the file size is greater than *zero*, then refer to the section [Backing Up the DSG Appliance OS from the Web UI](#) to ensure that all the audit logs are successfully sent to the Forensics before the DSG upgrade.

For more information about sending the audit logs to the Forensics, refer to the section [Pushing Audit Logs to Forensics](#).

- For each DSG node in the cluster, ensure that the `pepserver.cfg` configurations are saved before reimaging the DSG. These configurations must be manually set after the DSG is reimaged.

For more information about downloading the PEP server configuration file, refer to the section [Managing the PEP Server Configuration File](#).

- It is recommended that the Ruleset definitions are configured on the ESA such that the same configuration can be pushed simultaneously to all the nodes in the cluster.
- It is also recommended that any import or export of the DSG configurations and Ruleset configurations are performed from the ESA Web UI.

Caution:

The export and import of the DSG configurations and Ruleset configurations to DSG v3.1.0.5 is supported from the DSG v2.4.2 and higher. Navigate to **Settings > Backup & Restore** to export/import the DSG configurations and Ruleset configurations.

- Note:** Ensure that you do not import any DSG Ruleset backups (`.zip`) from the older DSG versions to the DSG 3.1.0.5.

- Caution:** If you are using codebook reshuffling, then ensure that for each DSG node where codebook reshuffling is configured, the following files are backed up before upgrading to the DSG 3.1.0.5:

- BLOB (random.dat)*
- dps.env*
- User PIN (userpin.bin)*

After backing up the files, you must create a `.tgz` package, which consists of all these files. Run the following command to create a `.tgz` package.

```
tar --same-owner -zcpvf /products/uploads/<filename>.tgz /opt/protegility/defiance_dps/data/random.dat /opt/protegility/defiance_dps/data/userpin.bin /opt/protegility/defiance_dps/bin/dps.env
```

Run the following command to set the required permissions for downloading the `.tgz` package from the DSG Web UI.

```
chmod 644 /products/uploads/<filename>.tgz
```

Perform the following steps to download the `.tgz` package from the DSG Web UI.

- Login to the DSG Web UI.



2. On the DSG Web UI, navigate to **Settings > System > File Upload**.
 3. Select the *.tgz* file from the *Uploaded Files* drop-down and click **Download**.
- Ensure that the *ESA_PAP-ALL-64_x86-64_9.1.0.5.xxxx.DSGUP.pty* patch is available. This patch is applied on the ESA to extend the ESA with the DSG Web UI.
 - Ensure that the *DSG_PAP-ALL-64_x86-64_3.1.0.5.x.UP.tgz* file is available. This file contains the *DSG_PAP-ALL-64_x86-64_3.1.0.5.x.UP.pty* and *PAP_PAP-ALL-64_x86-64_9.1.0.5.x.UP.pty* patch files.

Caution:

Ensure that you do not uninstall any Cloud Gateway service from the ESA.

Upgrading to the DSG v3.1.0.5 on ESA Node

Ensure that you use the following upgrade order to install the DSG v3.1.0.5 patch.

Table 9-3: Order of Upgrade for the ESA

Order of upgrade	Description	Reference
1	Upgrade ESA 9.1.0.3 to ESA 9.1.0.5.	Section Upgrading to v9.1.0.x in the <i>Protegility Upgrade Guide 9.1.0.5</i> .
2	Apply the DSG v3.1.0.5 patch (<i>ESA_PAP-ALL-64_x86-64_9.1.0.5.xxxx.DSGUP.pty</i>) on the ESA v9.1.0.5.	<i>Applying the DSG 3.1.0.5 Patch</i>

Upgrading to the DSG v3.1.0.5 from the DSG v3.1.0.3 Node

Ensure that you use the following table to upgrade the DSG nodes to the DSG v3.1.0.5.

Table 9-4: Order of Upgrade for the DSG

Order of installation	Description	Reference
1	Apply the <i>PAP_PAP-ALL-64_x86-64_9.1.0.5.x.UP.pty</i> Appliance Framework patch.	<i>Applying the Appliance Framework patch</i>
2	Apply the <i>DSG_PAP-ALL-64_x86-64_3.1.0.5.x.UP.pty</i> patch on the DSG 3.1.0.3.	<i>Applying the DSG 3.1.0.5 Patch</i>

Caution: If you are using codebook reshuffling, then ensure that the following codebook reshuffling configuration files for each DSG node are restored to the respective DSG node after upgrading to DSG 3.1.0.5:

- *BLOB (random.dat)*
- *dps.env*
- *User PIN (userpin.bin)*

For more information about configuring the backed up codebook reshuffling configuration, refer to the section *Restoring the Backed Up Codebook Reshuffling Configuration files*.

Note:

Ensure that the PEP server configurations on each DSG node are modified manually according to the *pepserver.cfg* file saved for the respective DSG node using the information in the section *Managing the PEP Server Configuration File*.

Note:

If any changes are made on a DSG node in the cluster, then create a scheduler task to replicate policies, configuration, DSG rulesets, and so on, from the DSG having changes to the other DSGs in the cluster.

For more information about creating a cluster task, refer to the section *Scheduling Configuration Export to Cluster Tasks* in the [Protegility Appliances Overview Guide 9.1.0.5](#).

Note:

If you are using UDFs in Rule definitions, then verify whether any of the blocked modules and methods are a part of your definitions.

For more information about handling UDFs in Rule definitions with blocked modules and methods, refer to the section [Verifying UDF Rules for Blocked Modules and Methods](#).

9.4 Upgrading to DSG v3.1.0.5 from DSG v3.1.0.2

This section describes the steps to upgrade to the DSG v3.1.0.5 from DSG v3.1.0.2.

Note:

You can upgrade from any version of DSG v3.1.0.2, including Hotfixes (HFs), Feature Enhancements (FEs), and Security Enhancements (SEs) to DSG v3.1.0.5.

Ensure that an ESA v9.1.0.2 is available with the DSG v3.1.0.2 patch installed on it.

Note:

For more information about upgrading the DSG component on the ESA, refer to the section [Upgrading to DSG v3.1.0.5](#).

Ensure that the DSG v3.1.0.2 is installed.

Note:

For more information about installing DSG v3.1.0.2 appliance, refer to the section *Installing the DSG* in the [Protegility Data Security Guide 3.1.0.4](#).

Before you begin:

Ensure that the following prerequisites are met before you upgrade the DSG to v3.1.0.5:

- Verify if the `pepservera<x>.dat` and `pepserverb<x>.dat` file size in the `/opt/protegility/defiance_dps/data/` directory is *zero*. If the file size is greater than *zero*, then refer to the section [Backing Up the DSG Appliance OS from the Web UI](#) to ensure that all the audit logs are successfully sent to the Forensics before the DSG upgrade.

For more information about sending the audit logs to the Forensics, refer to the section [Pushing Audit Logs to Forensics](#).

- For each DSG node in the cluster, ensure that the `pepserver.cfg` configurations are saved before reimaging the DSG. These configurations must be manually set after the DSG is reimaged.

For more information about downloading the PEP server configuration file, refer to the section [Managing the PEP Server Configuration File](#).

- It is recommended that the Ruleset definitions are configured on the ESA such that the same configuration can be pushed simultaneously to all the nodes in the cluster.



- It is also recommended that any import or export of the DSG configurations and Ruleset configurations are performed from the ESA Web UI.

Caution:

The export and import of the DSG configurations and Ruleset configurations to DSG v3.1.0.5 is supported from the DSG v2.4.2 and higher. Navigate to **Settings > Backup & Restore** to export/import the DSG configurations and Ruleset configurations.

- Note:** Ensure that you do not import any DSG Ruleset backups (.zip) from the older DSG versions to the DSG 3.1.0.5.

- Caution:** If you are using codebook reshuffling, then ensure that for each DSG node where codebook reshuffling is configured, the following files are backed up before upgrading to the DSG 3.1.0.5:

- BLOB (random.dat)*
- dps.env*
- User PIN (userpin.bin)*

After backing up the files, you must create a *.tgz* package, which consists of all these files. Run the following command to create a *.tgz* package.

```
tar --same-owner -zcpvf /products/
uploads/<filename>.tgz /opt/protegility/defiance_dps/data/random.dat /opt/protegility/
defiance_dps/data/userpin.bin /opt/protegility/defiance_dps/bin/dps.env
```

Run the following command to set the required permissions for downloading the *.tgz* package from the DSG Web UI.

```
chmod 644 /products/uploads/<filename>.tgz
```

Perform the following steps to download the *.tgz* package from the DSG Web UI.

- Login to the DSG Web UI.
- On the DSG Web UI, navigate to **Settings > System > File Upload**.
- Select the *.tgz* file from the *Uploaded Files* drop-down and click **Download**.

- Ensure that the *ESA_PAP-ALL-64_x86-64_9.1.0.5.xxxx.DSGUP.pty* patch is available. This patch is applied on the ESA to extend the ESA with the DSG Web UI.
- Ensure that the *DSG_PAP-ALL-64_x86-64_3.1.0.5.x.UP.tgz* file is available. This file contains the *DSG_PAP-ALL-64_x86-64_3.1.0.5.x.UP.pty* and *PAP_PAP-ALL-64_x86-64_9.1.0.5.x.UP.pty* patch files.

Caution:

Ensure that you do not uninstall any Cloud Gateway service from the ESA.

Upgrading to the DSG v3.1.0.5 on ESA Node

Ensure that you use the following upgrade order to install the DSG v3.1.0.5 patch.

Table 9-5: Order of Upgrade for the ESA

Order of upgrade	Description	Reference
1	Upgrade ESA 9.1.0.2 to ESA 9.1.0.5.	Section Upgrading to v9.1.0.x in the <i>Protegility Upgrade Guide 9.1.0.5</i> .

Order of upgrade	Description	Reference
2	Apply the DSG v3.1.0.5 patch (<i>ESA_PAP-ALL-64_x86-64_9.1.0.5.xxxx.DSGUP.pty</i>) on the ESA v9.1.0.5.	Applying the DSG 3.1.0.5 Patch

Upgrading to the DSG v3.1.0.5 from the DSG v3.1.0.2 Node

Ensure that you use the following table to upgrade the DSG nodes to the DSG v3.1.0.5.

Table 9-6: Order of Upgrade for the DSG

Order of installation	Description	Reference
1	Apply the <i>PAP_PAP-ALL-64_x86-64_9.1.0.5.x.UP.pty</i> Appliance Framework patch.	Applying the Appliance Framework patch
2	Apply the <i>DSG_PAP-ALL-64_x86-64_3.1.0.5.x.UP.pty</i> patch on the DSG 3.1.0.2.	Applying the DSG 3.1.0.5 Patch

Caution: If you are using codebook reshuffling, then ensure that the following codebook reshuffling configuration files for each DSG node are restored to the respective DSG node after upgrading to DSG 3.1.0.5:

- *BLOB (random.dat)*
- *dps.env*
- *User PIN (userpin.bin)*

For more information about configuring the backed up codebook reshuffling configuration, refer to the section [Restoring the Backed Up Codebook Reshuffling Configuration files](#).

Note:

Ensure that the PEP server configurations on each DSG node are modified manually according to the *pepserver.cfg* file saved for the respective DSG node using the information in the section [Managing the PEP Server Configuration File](#).

Note:

If any changes are made on a DSG node in the cluster, then create a scheduler task to replicate policies, configuration, DSG rulesets, and so on, from the DSG having changes to the other DSGs in the cluster.

For more information about creating a cluster task, refer to the section *Scheduling Configuration Export to Cluster Tasks* in the [Protegility Appliances Overview Guide 9.1.0.5](#).

Note:

If you are using UDFs in Rule definitions, then verify whether any of the blocked modules and methods are a part of your definitions.

For more information about handling UDFs in Rule definitions with blocked modules and methods, refer to the section [Verifying UDF Rules for Blocked Modules and Methods](#).

9.5 Upgrading to DSG v3.1.0.5 from DSG v3.1.0.1

This section describes the steps to upgrade to the DSG v3.1.0.5 from DSG v3.1.0.1.

Note:



You can upgrade from any version of DSG v3.1.0.1, including Hotfixes (HFs), Feature Enhancements (FEs), and Security Enhancements (SEs) to DSG v3.1.0.5.

Ensure that an ESA v9.1.0.1 is available with the DSG v3.1.0.1 patch installed on it.

Note:

For more information about upgrading the DSG component on the ESA, refer to the section [Upgrading to DSG v3.1.0.5](#).

Ensure that the DSG v3.1.0.1 is installed.

Note:

For more information about installing DSG v3.1.0.1 appliance, refer to the section *Installing the DSG* in the [Protegility Data Security Guide 3.1.0.1](#).

Before you begin:

Ensure that the following prerequisites are met before you upgrade the DSG to v3.1.0.5:

- Verify if the `pepservera<x>.dat` and `pepserverb<x>.dat` file size in the `/opt/protegility/defiance_dps/data/` directory is *zero*. If the file size is greater than *zero*, then refer to the section [Backing Up the DSG Appliance OS from the Web UI](#) to ensure that all the audit logs are successfully sent to the Forensics before the DSG upgrade.

For more information about sending the audit logs to the Forensics, refer to the section [Pushing Audit Logs to Forensics](#).

- For each DSG node in the cluster, ensure that the `pepserver.cfg` configurations are saved before reimaging the DSG. These configurations must be manually set after the DSG is reimaged.

For more information about downloading the PEP server configuration file, refer to the section [Managing the PEP Server Configuration File](#).

- It is recommended that the Ruleset definitions are configured on the ESA such that the same configuration can be pushed simultaneously to all the nodes in the cluster.
- It is also recommended that any import or export of the DSG configurations and Ruleset configurations are performed from the ESA Web UI.

Caution:

The export and import of the DSG configurations and Ruleset configurations to DSG v3.1.0.5 is supported from the DSG v2.4.2 and higher. Navigate to **Settings > Backup & Restore** to export/import the DSG configurations and Ruleset configurations.

- Note:** Ensure that you do not import any DSG Ruleset backups (.zip) from the older DSG versions to the DSG 3.1.0.5.

- Caution:** If you are using codebook reshuffling, then ensure that for each DSG node where codebook reshuffling is configured, the following files are backed up before upgrading to the DSG 3.1.0.5:
 - BLOB (random.dat)*
 - dps.env*
 - User PIN (userpin.bin)*



After backing up the files, you must create a *.tgz* package, which consists of all these files. Run the following command to create a *.tgz* package.

```
tar --same-owner -zcpvf /products/uploads/<filename>.tgz /opt/protegility/defiance_dps/data/random.dat /opt/protegility/defiance_dps/data/userpin.bin /opt/protegility/defiance_dps/bin/dps.env
```

Run the following command to set the required permissions for downloading the *.tgz* package from the DSG Web UI.

```
chmod 644 /products/uploads/<filename>.tgz
```

Perform the following steps to download the *.tgz* package from the DSG Web UI.

1. Login to the DSG Web UI.
2. On the DSG Web UI, navigate to **Settings > System > File Upload**.
3. Select the *.tgz* file from the *Uploaded Files* drop-down and click **Download**.

- Ensure that the *ESA_PAP-ALL-64_x86-64_9.1.0.5.xxxx.DSGUP.pty* patch is available. This patch is applied on the ESA to extend the ESA with the DSG Web UI.
- Ensure that the *DSG_PAP-ALL-64_x86-64_3.1.0.5.x.UP.tgz* file is available. This file contains the *DSG_PAP-ALL-64_x86-64_3.1.0.5.x.UP.pty* and *PAP_PAP-ALL-64_x86-64_9.1.0.5.x.UP.pty* patch files.

Caution:

Ensure that you do not uninstall any Cloud Gateway service from the ESA.

Upgrading to the DSG v3.1.0.5 on ESA Node

Ensure that you use the following upgrade order to install the DSG v3.1.0.5 patch.

Table 9-7: Order of Upgrade for the ESA

Order of upgrade	Description	Reference
1	Upgrade ESA 9.1.0.1 to ESA 9.1.0.5.	Section Upgrading to v9.1.0.x in the <i>Protegility Upgrade Guide 9.1.0.5</i> .
2	Apply the DSG v3.1.0.5 patch (<i>ESA_PAP-ALL-64_x86-64_9.1.0.5.xxxx.DSGUP.pty</i>) on the ESA v9.1.0.5. Note: Ensure that any patch application is performed using the Integrated Lights-Out (iLO) interface/VM console. Do not install this patch using an SSH connection.	<i>Extending ESA with DSG Web UI</i>

Upgrading to the DSG v3.1.0.5 from the DSG v3.1.0.1 Node

Ensure that you use the following table to upgrade the DSG nodes to the DSG v3.1.0.5.

Table 9-8: Order of Upgrade for the DSG

Order of installation	Description	Reference
1	Apply the <i>PAP_PAP-ALL-64_x86-64_9.1.0.5.x.UP.pty</i> Appliance Framework patch.	<i>Applying the Appliance Framework patch</i>

Order of installation	Description	Reference
2	Apply the <i>DSG_PAP-ALL-64_x86-64_3.1.0.5.x.UPpty</i> patch on the DSG 3.1.0.1.	Applying the DSG v3.1.0.5 patch

Caution: If you are using codebook reshuffling, then ensure that the following codebook reshuffling configuration files for each DSG node are restored to the respective DSG node after upgrading to DSG 3.1.0.5:

- *BLOB (random.dat)*
- *dps.env*
- *User PIN (userpin.bin)*

For more information about configuring the backed up codebook reshuffling configuration, refer to the section [Restoring the Backed Up Codebook Reshuffling Configuration files](#).

Note:

Ensure that the PEP server configurations on each DSG node are modified manually according to the *pepserver.cfg* file saved for the respective DSG node using the information in the section [Managing the PEP Server Configuration File](#).

Note:

If any changes are made on a DSG node in the cluster, then create a scheduler task to replicate policies, configuration, DSG rulesets, and so on, from the DSG having changes to the other DSGs in the cluster.

For more information about creating a cluster task, refer to the section *Scheduling Configuration Export to Cluster Tasks* in the [Protegility Appliances Overview Guide 9.1.0.5](#).

Note:

If you are using UDFs in Rule definitions, then verify whether any of the blocked modules and methods are a part of your definitions.

For more information about handling UDFs in Rule definitions with blocked modules and methods, refer to the section [Verifying UDF Rules for Blocked Modules and Methods](#).

9.6 Upgrading to DSG v3.1.0.5 from DSG v3.1.0.0

This section describes the steps to upgrade to the DSG v3.1.0.5 from DSG v3.1.0.0.

Note:

You can upgrade from any version of DSG v3.1.0.0, including Hotfixes (HFs), Feature Enhancements (FEs), and Security Enhancements (SEs) to DSG v3.1.0.5.

Ensure that an ESA v9.1.0.0 is available with the DSG v3.1.0.0 patch installed on it.

Note:

For more information about upgrading the DSG component on the ESA, refer to the section [Upgrading to DSG v3.1.0.5](#).

Ensure that the DSG v3.1.0.0 is installed.

Note:

For more information about installing DSG v3.1.0.0 appliance, refer to the section *Installing the DSG* in the *Protegility Data Security Guide 3.1.0.0*.

Before you begin:

Ensure that the following prerequisites are met before you upgrade the DSG to v3.1.0.5:

- Verify if the `pepservera<x>.dat` and `pepserverb<x>.dat` file size in the `/opt/protegility/defiance_dps/data/` directory is *zero*. If the file size is greater than *zero*, then refer to the section [Backing Up the DSG Appliance OS from the Web UI](#) to ensure that all the audit logs are successfully sent to the Forensics before the DSG upgrade.

For more information about sending the audit logs to the Forensics, refer to the section [Pushing Audit Logs to Forensics](#).

- For each DSG node in the cluster, ensure that the `pepserver.cfg` configurations are saved before reimaging the DSG. These configurations must be manually set after the DSG is reimaged.

For more information about downloading the PEP server configuration file, refer to the section [Managing the PEP Server Configuration File](#).

- It is recommended that the Ruleset definitions are configured on the ESA such that the same configuration can be pushed simultaneously to all the nodes in the cluster.
- It is also recommended that any import or export of the DSG configurations and Ruleset configurations are performed from the ESA Web UI.

Caution:

The export and import of the DSG configurations and Ruleset configurations to DSG v3.1.0.5 is supported from the DSG v2.4.2 and higher. Navigate to **Settings > Backup & Restore** to export/import the DSG configurations and Ruleset configurations.

- Note:** Ensure that you do not import any DSG Ruleset backups (.zip) from the older DSG versions to the DSG 3.1.0.5.

- Caution:** If you are using codebook reshuffling, then ensure that for each DSG node where codebook reshuffling is configured, the following files are backed up before upgrading to the DSG 3.1.0.5:

- BLOB (random.dat)*
- dps.env*
- User PIN (userpin.bin)*

After backing up the files, you must create a `.tgz` package, which consists of all these files. Run the following command to create a `.tgz` package.

```
tar --same-owner -zcpvf /products/uploads/<filename>.tgz /opt/protegility/defiance_dps/data/random.dat /opt/protegility/defiance_dps/data/userpin.bin /opt/protegility/defiance_dps/bin/dps.env
```

Run the following command to set the required permissions for downloading the `.tgz` package from the DSG Web UI.

```
chmod 644 /products/uploads/<filename>.tgz
```

Perform the following steps to download the `.tgz` package from the DSG Web UI.

- Login to the DSG Web UI.



2. On the DSG Web UI, navigate to **Settings > System > File Upload**.
 3. Select the *.tgz* file from the *Uploaded Files* drop-down and click **Download**.
- Ensure that the *ESA_PAP-ALL-64_x86-64_9.1.0.5.xxxx.DSGUP.pty* patch is available. This patch is applied on the ESA to extend the ESA with the DSG Web UI.
 - Ensure that the *DSG_PAP-ALL-64_x86-64_3.1.0.5.x.UP.tgz* file is available. This file contains the *DSG_PAP-ALL-64_x86-64_3.1.0.5.x.UP.pty* and *PAP_PAP-ALL-64_x86-64_9.1.0.5.x.UP.pty* patch files.

Caution:

Ensure that you do not uninstall any Cloud Gateway service from the ESA.

Upgrading to the DSG v3.1.0.5 on ESA Node

Ensure that you use the following upgrade order to install the DSG v3.1.0.5 patch.

Table 9-9: Order of Upgrade for the ESA

Order of upgrade	Description	Reference
1	Upgrade ESA 9.1.0.0 to ESA 9.1.0.5.	<i>Protegility Upgrade Guide 9.1.0.5</i>
2	Apply the DSG v3.1.0.5 patch (<i>ESA_PAP-ALL-64_x86-64_9.1.0.5.xxxx.DSGUP.pty</i>) on the ESA v9.1.0.5. Note: Ensure that any patch application is performed using the Integrated Lights-Out (iLO) interface/VM console. Do not install this patch using an SSH connection.	<i>Extending ESA with DSG Web UI</i>

Upgrading to the DSG v3.1.0.5 from the DSG v3.1.0.0 Node

Ensure that you use the following table to upgrade the DSG nodes to the DSG v3.1.0.5

Table 9-10: Order of Upgrade for the DSG

Order of installation	Description	Reference
1	Apply the <i>PAP_PAP-ALL-64_x86-64_9.1.0.5.x.UP.pty</i> Appliance Framework patch.	<i>Applying the Appliance Framework patch</i>
2	Apply the <i>DSG_PAP-ALL-64_x86-64_3.1.0.5.x.UP.pty</i> patch on the DSG 3.1.0.0.	<i>Applying the DSG v3.1.0.5 patch</i>

Caution: If you are using codebook reshuffling, then ensure that the following codebook reshuffling configuration files for each DSG node are restored to the respective DSG node after upgrading to DSG 3.1.0.5:

- *BLOB (random.dat)*
- *dps.env*
- *User PIN (userpin.bin)*

For more information about configuring the backed up codebook reshuffling configuration, refer to the section *Restoring the Backed Up Codebook Reshuffling Configuration files*.

Note:

Ensure that the PEP server configurations on each DSG node are modified manually according to the [`pepserver.cfg`](#) file saved for the respective DSG node using the information in the section [Managing the PEP Server Configuration File](#).

Note:

If any changes are made on a DSG node in the cluster, then create a scheduler task to replicate policies, configuration, DSG rulesets, and so on, from the DSG having changes to the other DSGs in the cluster.

For more information about creating a cluster task, refer to the section *Scheduling Configuration Export to Cluster Tasks* in the [Protegility Appliances Overview Guide 9.1.0.5](#).

Note:

If you are using UDFs in Rule definitions, then verify whether any of the blocked modules and methods are a part of your definitions.

For more information about handling UDFs in Rule definitions with blocked modules and methods, refer to the section [Verifying UDF Rules for Blocked Modules and Methods](#).

9.7 Upgrading to DSG v3.1.0.5 from DSG v3.0.0.0

This section describes the steps to upgrade to the DSG v3.1.0.5 from DSG v3.0.0.0.

Note:

You can upgrade from any version of DSG v3.0.0.0, including Hotfixes (HFs), Feature Enhancements (FEs), and Security Enhancements (SEs) to DSG v3.1.0.5.

Ensure that an ESA v9.0.0.0 is available with the DSG v3.1.0.0 patch installed on it.

Note:

For more information about upgrading the DSG component on the ESA, refer to the section [Upgrading to DSG v3.1.0.5](#).

Ensure that the DSG v3.0.0.0 is installed.

Note:

For more information about installing DSG v3.0.0.0 appliance, refer to the section *Installing the DSG* in the [Protegility Data Security Guide 3.0.0](#).

Before you begin:

Ensure that the following prerequisites are met before you upgrade the DSG to v3.1.0.5:

- Verify if the `pepservera<x>.dat` and `pepserverb<x>.dat` file size in the `/opt/protegility/defiance_dps/data/` directory is *zero*. If the file size is greater than *zero*, then refer to the section [Backing Up the DSG Appliance OS from the Web UI](#) to ensure that all the audit logs are successfully sent to the Forensics before the DSG upgrade.

For more information about sending the audit logs to the Forensics, refer to the section [Pushing Audit Logs to Forensics](#).

- For each DSG node in the cluster, ensure that the *pepserver.cfg* configurations are saved before reimaging the DSG. These configurations must be manually set after the DSG is reimaged.

For more information about downloading the PEP server configuration file, refer to the section [Managing the PEP Server Configuration File](#).

- It is recommended that the Ruleset definitions are configured on the ESA such that the same configuration can be pushed simultaneously to all the nodes in the cluster.
- It is also recommended that any import or export of the DSG configurations and Ruleset configurations are performed from the ESA Web UI.

Caution:

The export and import of the DSG configurations and Ruleset configurations to DSG v3.1.0.5 is supported from the DSG v2.4.2 and higher. Navigate to **Settings > Backup & Restore** to export/import the DSG configurations and Ruleset configurations.

- Note:** Ensure that you do not import any DSG Ruleset backups (.zip) from the older DSG versions to the DSG 3.1.0.5.

- Caution:** If you are using codebook reshuffling, then ensure that for each DSG node where codebook reshuffling is configured, the following files are backed up before upgrading to the DSG 3.1.0.5:

- BLOB (random.dat)*
- dps.env*
- User PIN (userpin.bin)*

After backing up the files, you must create a *.tgz* package, which consists of all these files. Run the following command to create a *.tgz* package.

```
tar --same-owner -zcpvf /products/
uploads/<filename>.tgz /opt/protegility/defiance_dps/data/random.dat /opt/protegility/
defiance_dps/data/userpin.bin /opt/protegility/defiance_dps/bin/dps.env
```

Run the following command to set the required permissions for downloading the *.tgz* package from the DSG Web UI.

```
chmod 644 /products/uploads/<filename>.tgz
```

Perform the following steps to download the *.tgz* package from the DSG Web UI.

1. Login to the DSG Web UI.
2. On the DSG Web UI, navigate to **Settings > System > File Upload**.
3. Select the *.tgz* file from the *Uploaded Files* drop-down and click **Download**.

- Ensure that the *ESA_PAP-ALL-64_x86-64_9.1.0.5.xxxx.DSGUP.pty* patch is available. This patch is applied on the ESA to extend the ESA with the DSG Web UI.
- Ensure that the *DSG_PAP-ALL-64_x86-64_3.1.0.5.x.UP-1.tgz* file is available. This file contains the *DSG_PAP-ALL-64_x86-64_3.1.0.5.x.UP.pty* and *PAP_PAP-ALL-64_x86-64_9.1.0.5.x.UP-1.pty* patch files.

Caution:

Ensure that you do not uninstall any Cloud Gateway service from the ESA.

Upgrading to the DSG v3.1.0.5 on ESA Node

Ensure that you use the following upgrade order to install the DSG v3.1.0.5 patch.

Table 9-11: Order of Upgrade for the ESA

Order of upgrade	Description	Reference
1	Upgrade ESA 9.0.0.0 to ESA 9.1.0.5.	Section Upgrading to v9.1.0.x in the <i>Protegility Upgrade Guide 9.1.0.5</i> .
2	Apply the DSG v3.1.0.5 patch (<i>ESA_PAP-ALL-64_x86-64_9.1.0.5.xxxx.DSGUP.pty</i>) on the ESA v9.1.0.5. Note: Ensure that any patch application is performed using the Integrated Lights-Out (iLO) interface/VM console. Do not install this patch using an SSH connection.	<i>Extending ESA with DSG Web UI</i>

Upgrading to the DSG v3.1.0.5 from the DSG v3.0.0.0 Node

Ensure that you use the following table to upgrade the DSG nodes to the DSG v3.1.0.5.

Table 9-12: Order of Upgrade for the DSG

Order of installation	Description	Reference
1	Apply the <i>PAP_PAP-ALL-64_x86-64_9.1.0.5.x.UP.pty</i> Appliance Framework patch.	<i>Applying the Appliance Framework patch</i>
2	Apply the <i>DSG_PAP-ALL-64_x86-64_3.1.0.5.x.UP.pty</i> patch on the DSG 3.0.0.0.	<i>Applying the DSG v3.1.0.5 patch</i>

Caution: If you are using codebook reshuffling, then ensure that the following codebook reshuffling configuration files for each DSG node are restored to the respective DSG node after upgrading to DSG 3.1.0.5:

- *BLOB (random.dat)*
- *dps.env*
- *User PIN (userpin.bin)*

For more information about configuring the backed up codebook reshuffling configuration, refer to the section *Restoring the Backed Up Codebook Reshuffling Configuration files*.

Note:

Ensure that the PEP server configurations on each DSG node are modified manually according to the *pepserver.cfg* file saved for the respective DSG node using the information in the section *Managing the PEP Server Configuration File*.

Note:

If any changes are made on a DSG node in the cluster, then create a scheduler task to replicate policies, configuration, DSG rulesets, and so on, from the DSG having changes to the other DSGs in the cluster.

For more information about creating a cluster task, refer to the section *Scheduling Configuration Export to Cluster Tasks* in the *Protegility Appliances Overview Guide 9.1.0.5*.

Note:

If you are using UDFs in Rule definitions, then verify whether any of the blocked modules and methods are a part of your definitions.

For more information about handling UDFs in Rule definitions with blocked modules and methods, refer to the section [Verifying UDF Rules for Blocked Modules and Methods](#).

9.8 Upgrading to DSG v3.1.0.5 from DSG v2.6.0.1

This section describes the steps to upgrade to the DSG v3.1.0.5 from the DSG v2.6.0.1.

Note:

You can upgrade from any version of DSG v2.6.0.1, including Hotfixes (HFs) and security vulnerabilities (SEs), to DSG v3.1.0.5.

Ensure that an ESA 8.1.0.1 is available with the DSG v2.6.0.1 patch installed on it.

Note:

For more information about upgrading the DSG component on the ESA, refer to the section [Upgrading to DSG v3.1.0.5](#).

Ensure that the DSG v2.6.0.1 is installed.

Note:

For more information about installing DSG v2.6.0.1 appliance, refer to the section *Installing the DSG* in the [Protegility Data Security Guide 2.6.0.1](#).

Before you begin:

Ensure that the following prerequisites are met before you upgrade the DSG to v3.1.0.5:

- Verify if the `pepservera<x>.dat` and `pepserverb<x>.dat` file size in the `/opt/protegility/defiance_dps/data/` directory is *zero*. If the file size is greater than *zero*, then refer to the section [Backing Up the DSG Appliance OS from the Web UI](#) to ensure that all the audit logs are successfully sent to the Forensics before the DSG upgrade.

For more information about sending the audit logs to the Forensics, refer to the section [Pushing Audit Logs to Forensics](#).

- For each DSG node in the cluster, ensure that the settings in the `pepserver.cfg` file are saved before reimaging the DSG. These configurations must be manually set after the DSG is reimaged.

For more information about downloading the PEP server configuration file, refer to the section [Managing the PEP Server Configuration File](#).

- It is recommended that the Ruleset definitions are configured on the ESA such that the same configuration can be pushed simultaneously to all the nodes in the cluster.
- It is also recommended that any import or export of the DSG configurations and Ruleset configurations are performed from the ESA Web UI.

Caution:

The export and import of the DSG configurations and Ruleset configurations to DSG v3.1.0.5 is supported from DSG v2.4.2 and higher.

- Note:** Ensure that you do not import any DSG Ruleset backups (.zip) from the older DSG versions to the DSG 3.1.0.5.



- **Caution:** If you are using codebook reshuffling, then ensure that for each DSG node where codebook reshuffling is configured, the following files are backed up before upgrading to the DSG 3.1.0.5:
 - *BLOB (random.dat)*
 - *dps.env*
 - *User PIN (userpin.bin)*

After backing up the files, you must create a *.tgz* package, which consists of all these files. Run the following command to create a *.tgz* package.

```
tar --same-owner -zcpvf /products/
uploads/<filename>.tgz /opt/protegility/defiance_dps/data/random.dat /opt/protegility/
defiance_dps/data/userpin.bin /opt/protegility/defiance_dps/bin/dps.env
```

Run the following command to set the required permissions for downloading the *.tgz* package from the DSG Web UI.

```
chmod 644 /products/uploads/<filename>.tgz
```

Perform the following steps to download the *.tgz* package from the DSG Web UI.

1. Login to the DSG Web UI.
2. On the DSG Web UI, navigate to **Settings > System > File Upload**.
3. Select the *.tgz* file from the *Uploaded Files* drop-down and click **Download**.

- Ensure that you backup the policies and rulesets related to the DSG from the ESA.
- Ensure that the *ESA_PAP-ALL-64_x86-64_9.1.0.5.xxxx.DSGUP.pty* patch is available. This patch is applied on the ESA to extend the ESA with the DSG Web UI.
- Ensure that the *DSG_PAP-ALL-64_x86-64_3.1.0.5.x.iso* is available. The *.iso* is used to install the DSG appliance.

Caution:

Ensure that you do not uninstall any Cloud Gateway service from the ESA.

Upgrading to the DSG v3.1.0.5 on ESA Node

Ensure that you use the following upgrade order to install the DSG v3.1.0.5 on the ESA.

Table 9-13: Order of Upgrade for the ESA

Order of upgrade	Description	Reference
1	Upgrade from ESA 8.1.0.1 to ESA 9.1.0.5.	Section Upgrading to v9.1.0.x in the <i>Protegility Upgrade Guide 9.1.0.5</i> .
2	Apply the <i>ESA_PAP-ALL-64_x86-64_9.1.0.5.xxxx.DSGUP.pty</i> on the ESA v9.1.0.5. This extends the ESA with DSG 3.1.0.5.	<i>Extending ESA with DSG Web UI</i>
	Note: Ensure that any patch application is performed using the Integrated Lights-Out (iLO) interface/VM console. Do not install this patch using an SSH connection.	
3	Import the ruleset configuration files on the new ESA.	
4	Create a cluster on the ESA.	<i>Creating a cluster</i>

Upgrading to the DSG v3.1.0.5 from the DSG v2.6.0.1 Node



Ensure that you use the following table to upgrade the DSG nodes to the DSG v3.1.0.5.

Table 9-14: Order of Upgrade for the DSG

Order of installation	Description	Reference
1	Reimage to the DSG v3.1.0.5.	Installing the DSG
2	Configure the DSG to forward the audit logs to <i>Audit Store</i> on the ESA.	Forwarding Logs to the Audit Store
3	Add the DSG to the ESA cluster.	Adding DSG to the cluster

Caution: If you are using codebook reshuffling, then ensure that the following codebook reshuffling configuration files for each DSG node are restored to the respective DSG node after upgrading to DSG 3.1.0.5:

- *BLOB (random.dat)*
- *dps.env*
- *User PIN (userpin.bin)*

For more information about configuring the backed up codebook reshuffling configuration, refer to the section [Restoring the Backed Up Codebook Reshuffling Configuration files](#).

Note:

Ensure that the PEP server configurations on each DSG node are modified manually according to the *pepperserver.cfg* file saved for the respective DSG node using the information in the section [Managing the PEP Server Configuration File](#).

Note:

If any changes are made on a DSG node in the cluster, then create a scheduler task to replicate policies, configuration, DSG rulesets, and so on, from the DSG having changes to the other DSGs in the cluster.

For more information about creating a cluster task, refer to the section *Scheduling Configuration Export to Cluster Tasks* in the [Protegility Appliances Overview Guide 9.1.0.5](#).

Note:

If your DSG contains UDF scripts that are written in Python 2, then they must be migrated to Python 3.

For more information about migrating scripts to Python 3, refer to section [Appendix I: Migrating the UDFs to Python 3](#).

Note:

If you are using UDFs in Rule definitions, then verify whether any of the blocked modules and methods are part of your definitions.

For more information about handling UDFs in Rule definitions with blocked modules and methods, refer to the section [Verifying UDF Rules for Blocked Modules and Methods](#).

9.9 Upgrading to DSG v3.1.0.5 from DSG v2.6.0.0

This section describes the steps to upgrade to the DSG v3.1.0.5 from the DSG v2.6.0.0.

Note:



You can upgrade from any version of the DSG v2.6.0.0, including Hotfixes (HFs) and security vulnerabilities (SEs), to DSG 3.1.0.5.

Ensure that an ESA 8.1.0.0 is available with the DSG v2.6.0.0 patch installed on it.

Note:

For more information about upgrading the DSG component on the ESA, refer to the section [Upgrading to DSG v3.1.0.5](#).

Ensure that the DSG v2.6.0.0 is installed.

Note:

For more information about installing DSG v2.6.0.0 appliance, refer to the section *Installing the DSG* in the [Protegility Data Security Guide 2.6.0](#).

Before you begin:

Ensure that the following prerequisites are met before you upgrade the DSG to v3.1.0.5:

- Verify if the `pepservera<x>.dat` and `pepserverb<x>.dat` file size in the `/opt/protegility/defiance_dps/data/` directory is *zero*. If the file size is greater than *zero*, then refer to the section [Backing Up the DSG Appliance OS from the Web UI](#) to ensure that all the audit logs are successfully sent to the Forensics before the DSG upgrade.
For more information about sending the audit logs to the Forensics, refer to the section [Pushing Audit Logs to Forensics](#).
- For each DSG node in the cluster, ensure that the settings in the `pepserver.cfg` file are saved before reimaging the DSG. These configurations must be manually set after the DSG is reimaged.
For more information about downloading the PEP server configuration file, refer to the section [Managing the PEP Server Configuration File](#).
- It is recommended that the Ruleset definitions are configured on the ESA such that the same configuration can be pushed simultaneously to all the nodes in the cluster.
- It is also recommended that any import or export of the DSG configurations and Ruleset configurations are performed from the ESA Web UI.

Caution:

The export and import of the DSG configurations and Ruleset configurations to DSG v3.1.0.5 is supported from DSG v2.4.2 and higher.

- Note:** Ensure that you do not import any DSG Ruleset backups (.zip) from the older DSG versions to the DSG 3.1.0.5.

- Caution:** If you are using codebook reshuffling, then ensure that for each DSG node where codebook reshuffling is configured, the following files are backed up before upgrading to the DSG 3.1.0.5:
 - BLOB (random.dat)*
 - dps.env*
 - User PIN (userpin.bin)*

After backing up the files, you must create a *.tgz* package, which consists of all these files. Run the following command to create a *.tgz* package.

```
tar --same-owner -zcpvf /products/uploads/<filename>.tgz /opt/protegility/defiance_dps/data/random.dat /opt/protegility/defiance_dps/data/userpin.bin /opt/protegility/defiance_dps/bin/dps.env
```

Run the following command to set the required permissions for downloading the *.tgz* package from the DSG Web UI.

```
chmod 644 /products/uploads/<filename>.tgz
```

Perform the following steps to download the *.tgz* package from the DSG Web UI.

1. Login to the DSG Web UI.
2. On the DSG Web UI, navigate to **Settings > System > File Upload**.
3. Select the *.tgz* file from the Uploaded Files drop-down and click **Download**.

- **Caution:** If you are using codebook reshuffling, then ensure that for each DSG node where codebook reshuffling is configured, the *BLOB* (*random.dat*) file and the *User PIN* (*userpin.bin*) file is backed up before reimaging to the DSG 3.1.0.5.

- Ensure that you backup the policies and rulesets related to the DSG from the ESA.
- Ensure that the *ESA_PAP-ALL-64_x86-64_9.1.0.5.xxxx.DSGUP.pty* patch is available. This patch is applied on the ESA to extend the ESA with the DSG Web UI.
- Ensure that the *DSG_PAP-ALL-64_x86-64_3.1.0.5.x.iso* is available. The *.iso* is used to install the DSG appliance.

Caution:

Ensure that you do not uninstall any Cloud Gateway service from the ESA.

Upgrading to the DSG v3.1.0.5 on ESA node

Ensure that you use the following upgrade order to install the DSG v3.1.0.5 on the ESA.

Table 9-15: Order of Upgrade for the ESA

Order of upgrade	Description	Reference
1	Upgrade from ESA 8.1.0.0 to ESA 9.1.0.5.	Section Upgrading to v9.1.0.x in the <i>Protegility Upgrade Guide 9.1.0.5</i> .
2	Apply the <i>ESA_PAP-ALL-64_x86-64_9.1.0.5.xxxx.DSGUP.pty</i> on the ESA v9.1.0.5. This extends the ESA with DSG 3.1.0.5.	<i>Extending ESA with DSG Web UI</i>
	<p>Note: Ensure that any patch application is performed using the Integrated Lights-Out (iLO) interface/VM console. Do not install this patch using an SSH connection.</p>	
3	Import the ruleset configuration files on the new ESA.	
4	Create a cluster on the ESA.	<i>Creating a cluster</i>

Upgrading to the DSG v3.1.0.5 from the DSG v2.6.0.0 Node

Ensure that you use the following table to upgrade the DSG nodes to the DSG v3.1.0.5.



Table 9-16: Order of Upgrade for the DSG

Order of installation	Description	Reference
1	Reimage to the DSG v3.1.0.5.	Installing the DSG
2	Configure the DSG to forward the audit logs to <i>Audit Store</i> on the ESA.	Forwarding Logs to the Audit Store
3	Add the DSG to the ESA cluster.	Adding DSG to the cluster

Note:

Ensure that the PEP server configurations on each DSG node are modified manually according to the `pepserver.cfg` file saved for the respective DSG node using the information in the section [Managing the PEP Server Configuration File](#).

Note:

If any changes are made on a DSG node in the cluster, then create a scheduler task to replicate policies, configuration, DSG rulesets, and so on, from the DSG having changes to the other DSGs in the cluster.

For more information about creating a cluster task, refer to the section *Scheduling Configuration Export to Cluster Tasks* in the [Protegility Appliances Overview Guide 9.1.0.5](#).

Note:

If you are using UDFs in Rule definitions, then verify whether any of the blocked modules and methods are part of your definitions.

For more information about handling UDFs in Rule definitions with blocked modules and methods, refer to the section [Verifying UDF Rules for Blocked Modules and Methods](#).

Note:

If your DSG contains UDF scripts that are written in Python 2, then they must be migrated to Python 3.

For more information about migrating scripts to Python 3, refer to section [Appendix I: Migrating the UDFs to Python 3](#).

9.10 Upgrading to DSG v3.1.0.5 from DSG v2.4.2

This section describes the steps to upgrade to the DSG v3.1.0.5 from the DSG v2.4.2.

Note:

You can upgrade from any version of the DSG v2.4.2, including Hotfixes (HFs) and security vulnerabilities (SEs), to the DSG v3.1.0.5.

Ensure that an ESA v7.2.1 is available with the DSG v2.4.2 patch installed on it.

Note:

For more information about upgrading the DSG component on the ESA, refer to the section [Upgrading to DSG v3.1.0.5](#).

Ensure that the DSG v2.4.2 is installed.

Note:

For more information about installing DSG v2.4.2 appliance, refer to the section *Installing the DSG* in the *Protegility Data Security Guide 2.4.2*.

Before you begin:

Ensure that the following prerequisites are met before you upgrade the DSG to v3.1.0.5:

- Verify if the `pepservera<x>.dat` and `pepserverb<x>.dat` file size in the `/opt/protegility/defiance_dps/data/` directory is *zero*. If the file size is greater than *zero*, then refer to the section [Backing Up the DSG Appliance OS from the Web UI](#) to ensure that all the audit logs are successfully sent to the Forensics before the DSG upgrade.
For more information about sending the audit logs to the Forensics, refer to the section [Pushing Audit Logs to Forensics](#).
- For each DSG node in the cluster, ensure that the settings in the `pepserver.cfg` file are saved before reimaging the DSG. These configurations must be manually set after the DSG is reimaged.
For more information about downloading the PEP server configuration file, refer to the section [Managing the PEP Server Configuration File](#).
- It is recommended that the Ruleset definitions are configured on the ESA such that the same configuration can be pushed simultaneously to all the nodes in the cluster.
- It is also recommended that any import or export of the DSG configurations and Ruleset configurations are performed from the ESA Web UI.

Caution:

The export and import of the DSG configurations and Ruleset configurations to the DSG v3.1.0.5 is supported from DSG v2.4.2 and higher.

- Note:** Ensure that you do not import any DSG Ruleset backups (.zip) from the older DSG versions to the DSG 3.1.0.5.

- Caution:** If you are using codebook reshuffling, then ensure that for each DSG node where codebook reshuffling is configured, the following files are backed up before upgrading to the DSG 3.1.0.5:
 - BLOB (random.dat)*
 - dps.env*
 - User PIN (userpin.bin)*

After backing up the files, you must create a `.tgz` package, which consists of all these files. Run the following command to create a `.tgz` package.

```
tar --same-owner -zcpvf /products/uploads/<filename>.tgz /opt/protegility/defiance_dps/data/random.dat /opt/protegility/defiance_dps/data/userpin.bin /opt/protegility/defiance_dps/bin/dps.env
```

Run the following command to set the required permissions for downloading the `.tgz` package from the DSG Web UI.

```
chmod 644 /products/uploads/<filename>.tgz
```

Perform the following steps to download the `.tgz` package from the DSG Web UI.

1. Login to the DSG Web UI.



2. On the DSG Web UI, navigate to **Settings > System > File Upload**.
3. Select the *.tgz* file from the Uploaded Files drop-down and click **Download**.

- Ensure that you backup the policies and rulesets related to the DSG from the ESA.
- Ensure that the *ESA_PAP-ALL-64_x86-64_9.1.0.5.xxxx.DSGUP.pty* patch is available. This patch is applied on the ESA to extend the ESA with the DSG Web UI.
- Ensure that the *DSG_PAP-ALL-64_x86-64_3.1.0.5.x.iso* is available. The *.iso* is used to install the DSG appliance.

Caution:

Ensure that you do not uninstall any Cloud Gateway service from the ESA.

Upgrading to the DSG v3.1.0.5 on ESA node

Ensure that you use the following upgrade order to install the DSG v3.1.0.5 on the ESA.

Table 9-17: Order of Upgrade for the ESA

Order of upgrade	Description	Reference
1	Upgrade from ESA 7.2.1 to ESA 9.1.0.5.	<i>Protegility Upgrade Guide 9.1.0.5</i>
2	Apply the <i>ESA_PAP-ALL-64_x86-64_9.1.0.5.xxxx.DSGUP.pty</i> on the ESA v9.1.0.5. This extends the ESA with DSG 3.1.0.5. Note: Ensure that any patch application is performed using the Integrated Lights-Out (iLO) interface/VM console. Do not install this patch using an SSH connection.	<i>Extending ESA with DSG Web UI</i>
3	Import the ruleset configuration files on the new ESA.	
4	Create a cluster on the ESA.	<i>Creating a cluster</i>

Upgrading to the DSG v3.1.0.5 from the DSG v2.4.2 Node

Ensure that you use the following table to upgrade the DSG nodes to the DSG v3.1.0.5.

Table 9-18: Order of Upgrade for the DSG

Order of installation	Description	Reference
1	Reimage to the DSG v3.1.0.5.	<i>Installing the DSG</i>
2	Configure the DSG to forward the audit logs to <i>Audit Store</i> on the ESA.	<i>Forwarding Logs to the Audit Store</i>
3	Add the DSG to the ESA cluster.	<i>Adding DSG to the cluster</i>

Caution: If you are using codebook reshuffling, then ensure that the following codebook reshuffling configuration files for each DSG node are restored to the respective DSG node after upgrading to DSG 3.1.0.5:

- *BLOB (random.dat)*
- *dps.env*
- *User PIN (userpin.bin)*

For more information about configuring the backed up codebook reshuffling configuration, refer to the section *Restoring the Backed Up Codebook Reshuffling Configuration files*.

Note:

Ensure that the PEP server configurations on each DSG node are modified manually according to the `pepserver.cfg` file saved for the respective DSG node using the information in the section [Managing the PEP Server Configuration File](#).

Note:

If any changes are made on a DSG node in the cluster, then create a scheduler task to replicate policies, configuration, DSG rulesets, and so on, from the DSG having changes to the other DSGs in the cluster.

For more information about creating a cluster task, refer to the section *Scheduling Configuration Export to Cluster Tasks* in the [Protegility Appliances Overview Guide 9.1.0.5](#).

Note:

If you are using UDFs in Rule definitions, then verify whether any of the blocked modules and methods are part of your definitions.

For more information about handling UDFs in Rule definitions with blocked modules and methods, refer to the section [Verifying UDF Rules for Blocked Modules and Methods](#).

Note:

If your DSG contains UDF scripts that are written in Python 2, then they must be migrated to Python 3.

For more information about migrating scripts to Python 3, refer to section [Appendix I: Migrating the UDFs to Python 3](#).

9.11 Upgrading to DSG v3.1.0.5 from DSG v2.4.1

This section describes the steps to upgrade to the DSG v3.1.0.5 from the DSG v2.4.1.

Ensure that an ESA v7.2.1 is available with the DSG v2.4.1 patch installed on it.

Note:

For more information about upgrading the DSG component on the ESA, refer to the section [Upgrading to DSG v3.1.0.5](#).

Ensure that the DSG v2.4.1 is installed.

Note:

For more information about installing DSG v2.4.1 appliance, refer to the section *Installing the DSG* in the [Protegility Data Security Guide 2.4.1](#).

Before you begin:

Ensure that the following prerequisites are met before you upgrade the DSG to v3.1.0.5:

- Verify if the `pepservera<x>.dat` and `pepserverb<x>.dat` file size in the `/opt/protegility/defiance_dps/data/` directory is *zero*. If the file size is greater than *zero*, then refer to the section [Backing Up the DSG Appliance OS from the Web UI](#) to ensure that all the audit logs are successfully sent to the Forensics before the DSG upgrade.

For more information about sending the audit logs to the Forensics, refer to the section [Pushing Audit Logs to Forensics](#).



- For each DSG node in the cluster, ensure that the settings in the *pepserver.cfg* file are saved before reimaging the DSG. These configurations must be manually set after the DSG is reimaged.

For more information about downloading the PEP server configuration file, refer to the section [Managing the PEP Server Configuration File](#).

- It is recommended that the Ruleset definitions are configured on the ESA such that the same configuration can be pushed simultaneously to all the nodes in the cluster.
- It is also recommended that any import or export of the DSG configurations and Ruleset configurations are performed from the ESA Web UI.

Caution:

The export and import of the DSG configurations and Ruleset configurations to the DSG v3.1.0.5 are supported from DSG v2.4.2 and higher.

- Note:** Ensure that you do not import any DSG Ruleset backups (.zip) from the older DSG versions to the DSG 3.1.0.5.

- Caution:** If you are using codebook reshuffling, then ensure that for each DSG node where codebook reshuffling is configured, the following files are backed up before upgrading to the DSG 3.1.0.5:

- BLOB (random.dat)*
- dps.env*
- User PIN (userpin.bin)*

After backing up the files, you must create a *.tgz* package, which consists of all these files. Run the following command to create a *.tgz* package.

```
tar --same-owner -zcpvf /products/
uploads/<filename>.tgz /opt/protegility/defiance_dps/data/random.dat /opt/protegility/
defiance_dps/data/userpin.bin /opt/protegility/defiance_dps/bin/dps.env
```

Run the following command to set the required permissions for downloading the *.tgz* package from the DSG Web UI.

```
chmod 644 /products/uploads/<filename>.tgz
```

Perform the following steps to download the *.tgz* package from the DSG Web UI.

1. Login to the DSG Web UI.
2. On the DSG Web UI, navigate to **Settings > System > File Upload**.
3. Select the *.tgz* file from the Uploaded Files drop-down and click **Download**.

- Caution:** If you are using codebook reshuffling, then ensure that for each DSG node where codebook reshuffling is configured, the *BLOB (random.dat)* file and the *User PIN (userpin.bin)* file is backed up before reimaging to the DSG 3.1.0.5.

- Ensure that you backup the policies and rulesets related to the DSG from the ESA.
- Ensure that the *ESA_PAP-ALL-64_x86-64_9.1.0.5.xxxx.DSGUP.pty* patch is available. This patch is applied on the ESA to extend the ESA with the DSG Web UI.
- Ensure that the *DSG_PAP-ALL-64_x86-64_3.1.0.5.x.iso* is available. The *.iso* is used to install the DSG appliance.

Caution:



Ensure that you do not uninstall any Cloud Gateway service from the ESA.

Upgrading to the DSG v3.1.0.5 from the DSG v2.4.1 Node

Ensure that you use the following upgrade order to install the DSG v3.1.0.5 on the ESA.

Table 9-19: Order of Upgrade for the ESA

Order of upgrade	Description	Reference
1	Upgrade from ESA 7.2.1 to ESA 9.1.0.5.	Section Upgrading to v9.1.0.x in the Protegility Upgrade Guide 9.1.0.5 .
2	Apply the <i>ESA_PAP-ALL-64_x86-64_9.1.0.5.xxxx.DSGUP.pty</i> on the ESA v9.1.0.5. This extends the ESA with DSG 3.1.0.5. Note: Ensure that any patch application is performed using the Integrated Lights-Out (iLO) interface/VM console. Do not install this patch using an SSH connection.	Extending ESA with DSG Web UI
3	Import the ruleset configuration files on the new ESA.	
4	Create a cluster on the ESA.	Creating a cluster

Upgrading to the DSG v3.1.0.5 from the DSG Node

Ensure that you use the following table to upgrade the DSG nodes to the DSG v3.1.0.5.

Table 9-20: Order of Upgrade for the DSG

Order of installation	Description	Reference
1	Reimage to the DSG v3.1.0.5.	Installing the DSG
2	Configure the DSG to forward the audit logs to <i>Audit Store</i> on the ESA.	Forwarding Logs to the Audit Store
3	Add the DSG to the ESA cluster.	Adding DSG to the cluster

Caution: After upgrading to the DSG 3.1.0.5, ensure that any backed up *BLOB (random.dat)* file and the *User PIN (userpin.bin)* file for each DSG node is restored to the respective DSG node.

Note:

Ensure that the PEP server configurations on each DSG node are modified manually according to the *pepserver.cfg* file saved for the respective DSG node using the information in the section [Managing the PEP Server Configuration File](#).

Note:

If any changes are made on a DSG node in the cluster, then create a scheduler task to replicate policies, configuration, DSG rulesets, and so on, from the DSG having changes to the other DSGs in the cluster.

For more information about creating a cluster task, refer to the section *Scheduling Configuration Export to Cluster Tasks* in the [Protegility Appliances Overview Guide 9.1.0.5](#).

Note:

If you are using UDFs in Rule definitions, then verify whether any of the blocked modules and methods are part of your definitions.



For more information about handling UDFs in Rule definitions with blocked modules and methods, refer to the section [Verifying UDF Rules for Blocked Modules and Methods](#).

Note:

If your DSG contains UDF scripts that are written in Python 2, then they must be migrated to Python 3.

For more information about migrating scripts to Python 3, refer to section [Appendix I: Migrating the UDFs to Python 3](#).

9.12 Upgrading to DSG v3.1.0.5 from DSG v2.4.0

This section describes the steps to upgrade to the DSG v3.1.0.5 from the DSG v2.4.0.

Ensure that an ESA v7.2.1 is available with the DSG v2.4.0 patch installed on it.

Note:

For more information about upgrading the DSG component on the ESA, refer to the section [Upgrading to DSG v3.1.0.5](#).

Ensure that the DSG v2.4.0 is installed.

Note:

For more information about installing DSG v2.4.0 appliance, refer to the section [Installing the DSG](#) in the [Protegility Data Security Guide 2.4.0](#).

Before you begin:

Ensure that the following prerequisites are met before you upgrade the DSG to v3.1.0.5:

- Verify if the `pepservera<x>.dat` and `pepserverb<x>.dat` file size in the `/opt/protegility/defiance_dps/data/` directory is *zero*. If the file size is greater than *zero*, then refer to the section [Backing Up the DSG Appliance OS from the Web UI](#) to ensure that all the audit logs are successfully sent to the Forensics before the DSG upgrade.
For more information about sending the audit logs to the Forensics, refer to the section [Pushing Audit Logs to Forensics](#).
- For each DSG node in the cluster, ensure that the settings in the `pepserver.cfg` file are saved before reimaging the DSG. These configurations must be manually set after the DSG is reimaged.
For more information about downloading the PEP server configuration file, refer to the section [Managing the PEP Server Configuration File](#).

- It is recommended that the Ruleset definitions are configured on the ESA such that the same configuration can be pushed simultaneously to all the nodes in the cluster.
- It is also recommended that any import or export of the DSG configurations and Ruleset configurations are performed from the ESA Web UI.

Caution:

The export and import of the DSG configurations and Ruleset configurations to the DSG v3.1.0.5 is supported from DSG v2.4.2 and higher.

- **Note:** Ensure that you do not import any DSG Ruleset backups (.zip) from the older DSG versions to the DSG 3.1.0.5.
- **Caution:** If you are using codebook reshuffling, then ensure that for each DSG node where codebook reshuffling is configured, the following files are backed up before upgrading to the DSG 3.1.0.5:
 - *BLOB (random.dat)*
 - *dps.env*
 - *User PIN (userpin.bin)*

After backing up the files, you must create a *.tgz* package, which consists of all these files. Run the following command to create a *.tgz* package.

```
tar --same-owner -zcpvf /products/uploads/<filename>.tgz /opt/protegility/defiance_dps/data/random.dat /opt/protegility/defiance_dps/data/userpin.bin /opt/protegility/defiance_dps/bin/dps.env
```

Run the following command to set the required permissions for downloading the *.tgz* package from the DSG Web UI.

```
chmod 644 /products/uploads/<filename>.tgz
```

Perform the following steps to download the *.tgz* package from the DSG Web UI.

1. Login to the DSG Web UI.
2. On the DSG Web UI, navigate to **Settings > System > File Upload**.
3. Select the *.tgz* file from the Uploaded Files drop-down and click **Download**.

- Ensure that you backup the policies and rulesets related to the DSG from the ESA.
- Ensure that the *ESA_PAP-ALL-64_x86-64_9.1.0.5.xxxx.DSGUP.pty* patch is available. This patch is applied on the ESA to extend the ESA with the DSG Web UI.
- Ensure that the *DSG_PAP-ALL-64_x86-64_3.1.0.5.x.iso* is available. The *.iso* is used to install the DSG appliance.

Caution:

Ensure that you do not uninstall any Cloud Gateway service from the ESA.

Upgrading to the DSG v3.1.0.5 on ESA Node

Ensure that you use the following upgrade order to install the DSG v3.1.0.5 on the ESA.

Table 9-21: Order of Upgrade for the ESA

Order of upgrade	Description	Reference
1	Upgrade from ESA 7.2.1 to ESA 9.1.0.5.	Section Upgrading to v9.1.0.x in the <i>Protegility Upgrade Guide 9.1.0.5</i> .
2	Apply the <i>ESA_PAP-ALL-64_x86-64_9.1.0.5.xxxx.DSGUP.pty</i> on the ESA v9.1.0.5. This extends the ESA with DSG 3.1.0.5.	<i>Extending ESA with DSG Web UI</i>
3	Note: Ensure that any patch application is performed using the Integrated Lights-Out (iLO) interface/VM console. Do not install this patch using an SSH connection.	



Order of upgrade	Description	Reference
4	Create a cluster on the ESA.	Creating a cluster

Upgrading to the DSG v3.1.0.5 from the DSG v2.4.0 Node

Ensure that you use the following table to reimage the DSG nodes to the DSG v3.1.0.5.

Table 9-22: Order of Upgrade for the DSG

Order of installation	Description	Reference
1	Reimage to the DSG v3.1.0.5.	Installing the DSG
2	Configure the DSG to forward the audit logs to <i>Audit Store</i> on the ESA.	Forwarding Logs to the Audit Store
3	Add the DSG to the ESA cluster.	Adding DSG to the cluster

Caution: After upgrading to the DSG 3.1.0.5, ensure that any backed up *BLOB* (*random.dat*) file and the *User PIN* (*userpin.bin*) file for each DSG node is restored to the respective DSG node.

Note:

Ensure that the PEP server configurations on each DSG node are modified manually according to the *pepserver.cfg* file saved for the respective DSG node using the information in the section [Managing the PEP Server Configuration File](#).

Note:

If any changes are made on a DSG node in the cluster, then create a scheduler task to replicate policies, configuration, DSG rulesets, and so on, from the DSG having changes to the other DSGs in the cluster.

For more information about creating a cluster task, refer to the section *Scheduling Configuration Export to Cluster Tasks* in the [Protegility Appliances Overview Guide 9.1.0.5](#).

Note:

If you are using UDFs in Rule definitions, then verify whether any of the blocked modules and methods are part of your definitions.

For more information about handling UDFs in Rule definitions with blocked modules and methods, refer to the section [Verifying UDF Rules for Blocked Modules and Methods](#).

Note:

If your DSG contains UDF scripts that are written in Python 2, then they must be migrated to Python 3.

For more information about migrating scripts to Python 3, refer to section [Appendix I: Migrating the UDFs to Python 3](#).

9.13 Restoring the DSG Appliance OS Backup

This section describes the steps to restore the DSG Appliance OS backup.



► To restore the DSG Appliance OS backup:

1. Login to the DSG CLI.
2. Navigate to **Administration > Reboot and Shutdown**.
3. Select **Reboot** and press **Enter**.
4. Provide a reason for restarting the DSG node, select **OK** and press **Enter**.
5. Enter the root password, select **OK** and press **Enter**.
The restart operation is initiated.
6. During start up, select the **System Restore Mode**.
7. Select **Initiate OS-Restore Procedure**.
8. Select **OK**.
The restore process is initiated.
After the restore process is completed, the login screen appears.

9.14 Pushing Audit Logs to Forensics

This section applies when you are upgrading from the older releases, such as, DSG 2.x or DSG 1.x, to the DSG 3.1.0.5. The steps in this section ensure that all audit logs are sent to the Forensics before the DSG node is upgraded.

Before you begin

Verify if the **pepservera<x>.dat** and **pepserverb<x>.dat** file size in the **/opt/protegility/defiance_dps/data/** directory is *zero*. If the file size is more than *zero*, then ensure that the *logfacade* service is restarted. This step ensures that all the audit logs are successfully sent to the Forensics before the DSG upgrade.

There can exist multiple files with the prefix **pepservera** and **pepserverb** with the extension **.dat** ranging from *1* to *x*. Ensure that file size of such **.dat** files is *zero*.

► To push audit logs to forensics:

1. Login to the ESA Web UI.
2. Navigate to **System > Services**.
3. Under the **Policy Management** area, stop the **Logfacade** service, and then start the **Logfacade** service.
4. Restart the *PEP server* service by performing the following steps.
 - a. Login to the DSG Web UI.
 - b. Navigate to **System > Services**.
 - c. Restart the **Pepserver Service**.
5. Verify that *.dat* files in the directory are approximately *0 MB* in size.
6. Repeat the step 4 and 5 on each DSG node in the cluster.

9.15 Managing the PEP Server Configuration File

This section applies when you are upgrading from the any older DSG version to the DSG v3.1.0.5.

► To download the PEP server configuration file:

1. Login to the DSG Web UI.
2. Navigate to **Settings > System**.
3. Under the **Files** tab, download the *pepserver.cfg* file.
4. Repeat step 1 to step 3 on each DSG node in the cluster.

9.16 Restarting the DSG Node

This section applies when an appliance framework patch is applied and the DSG node must be restarted.

► To restarted the DSG node:

1. On the DSG CLI Manager, navigate to **Administration > Reboot and Shutdown**.
2. Select **Reboot** and press **Enter**.
3. Provide a reason for restarting the DSG node, select **OK** and press **Enter**.
4. Enter the **root** password, select **OK** and press **Enter**.

The DSG node is restarted.

9.17 Restoring LogForwarder Custom Files

In DSG, the custom files related to LogForwarder can be placed in the */opt/protegility/fluent-bit/data/config.d* directory. These files can be configured as required. When the upgrade is in process, DSG backs up the custom LogForwarder files. The files are moved to the */opt/protegility/fluent_backup* directory.

However, after the upgrade is completed, the files are not automatically restored to the */opt/protegility/fluent-bit/data/config.d* directory. All the custom configuration files and the modifications to existing default configuration files done before upgrade must to be manually applied after upgrade.

9.18 Restoring the Backed Up Codebook Reshuffling Configuration files

This section describes the steps to restore the backed up Codebook Reshuffling configuration files after upgrading the DSG.

Note: The Codebook Reshuffling feature is tested and supported for the Safenet Luna 7.4 HSM device. The procedure provided in this section is for the Safenet Luna 7.4 HSM device.

Before you begin

It is recommended to configure the HSM before restoring the backed up codebook reshuffling configuration files.

► To restore the backed up Codebook Reshuffling configuration files:

1. Login to the DSG Web UI.
2. On the DSG Web UI, navigate to **Settings > System > File Upload**.

Note: By default, the *Max File Upload* size is set to **25 MB** on the DSG appliances. If the *<filename>.tgz* file size is more than 25 MB, the *Max File Upload* size must be changed. If this value is set to 2 GB, then the following steps can be ignored.

Perform the following steps to increase the *Max File Upload* size:

1. On the DSG Web UI, navigate to **Settings > Network > Web Settings**.
2. Under General Settings, ensure that the *Max File Upload* is set to 2 GB to accommodate the patch upload.
3. Ensure that the steps 1 and 2 are performed on each DSG node in the cluster.

3. On the File Selection screen, select the *<filename>.tgz* file, which consists of the following backed up codebook reshuffling files, and click **Upload**:
 - *BLOB (random.dat)*
 - *dps.env*
 - *User PIN (userpin.bin)*
4. Login to the DSG CLI Manager.
5. Navigate to **Administration > OS Console**.
6. Enter the root password.
7. Navigate to the */products/uploads* directory by running the following command.

```
cd /products/uploads
```

8. Run the following command to extract the contents of the *<filename>.tgz* file.

```
tar -xvpf <filename>.tgz -C /
```

The contents of the *<filename>.tgz* file are extracted.

9. Setup the Token Domain for Codebook Reshuffling by running the following commands.

```
cd /opt/protegility/defiance_dps/data
su -s /bin/sh service_admin -c "ln -s /opt/protegility/hsm/libCryptoki2_64.so pkcs11.plm"
```

10. Run the following command to source the *dps.env* file.

```
. /opt/protegility/defiance_dps/bin/dps.env
```

Note: The command has a dot followed by a space and then the path.

11. Ensure that you have set the *shufflecodebooks* configuration parameter to *yes* and the path to the file containing the random bytes in the *pepper.cfg* configuration file using the following code snippet.

```
# shuffle token codebooks after they are downloaded.
# yes, no. default no.

shufflecodebooks = yes
# Path to the file that contains the random bytes for shuffling codebooks.
randomfile = ./random.dat
```

12. Ensure that you have set the required path to the *PKCS#11* provider library, slot number to be used on the HSM, and the required path to the *userpin.bin* file in the *pepper.cfg* configuration file using the following code snippet.

```
# -----
# PKCS#11 configuration
# Values in this section is only used
```

```
# when shufflecodebooks = yes
# -----
[pkcs11]

# The path to the PKCS#11 provider library.
provider_library = ./pkcs11.pbm

# The slot number to use on the HSM.
slot = 1

# The scrambled user pin file.
userpin = ./userpin.bin
```

Note: The *PKCS#11* configuration parameter is available in the *PKCS#11 configuration* section of the *pepserver.cfg* file.

13. On the DSG Web UI, navigate to **System > Services** to restart the PEP server.

The backed up Codebook Reshuffling configuration files are restored.

9.19 Verifying UDF Rules for Blocked Modules and Methods

If you are using UDFs in Rule definitions, then it is important to verify whether you are using any of the blocked modules and methods. This section provides information about handling such blocked modules or methods. The introduction of blocking is a security best practice that restricts UDF code instructions to use safe modules and methods.

After installing the DSG, ensure that you note the following points:

- Verify if any of the following blocked modules and methods are defined in the *Source Code* option in the UDF rules:
 - **blocked_modules:** pip , install, commands, subprocess, popen2, sys, os, platform, signal, asyncio
 - **blocked_methods:** eval, exec, dir, import, memoryview
- If any of the blocked modules or methods are defined in the *Source Code* option in the UDF rules, then use either of the following options:
 - **Option 1:** Remove the module/method from the *gateway.json* file.

For more information about editing the *gateway.json*, refer to the section [Blocked Modules and Methods in UDF](#).

Note:

By removing blocked modules and methods, you risk introducing security risks to the DSG system should any UDF code misuse these otherwise blocked module/method.

- **Option 2:** Edit the UDF rule to override the blocked module using the *override_blocked_modules* parameter.

For more information about editing the *override_blocked_modules* parameter, refer to the section [Advanced Rule Settings in UDFs](#).

Note:

By overriding blocked modules, you risk introducing security risks to the DSG system should any UDF code misuse these otherwise blocked module.

9.20 Products Compatibility Matrix

This section contains the details for the compatibility matrix between various ESA and DSG versions.

DSG	Compatible ESA
2.4.0	7.2.1
2.4.1, 2.4.1 HF-1, 2.4.1 SE-1, 2.4.1 FE-2, 2.4.2	7.2.1
2.5.0.0	8.0.0.0
2.6.0.0, 2.6.0.0 HF-1, 2.6.0.0 SE-1, 2.6.0.0 HF-3, 2.6.0.0 SE-2, 2.6.0.0 HF-4	8.1.0.0
2.6.0.1, 2.6.0.1 HF-1, 2.6.0.1 SE-1, 2.6.0.1 HF-2, 2.6.0.1 SE-2, 2.6.0.1 HF-4	8.1.0.1
3.0.0.0, 3.0.0.0 HF-1, 3.0.0.0 HF-2, 3.0.0.0 SE-2, 3.0.0.0 HF-3	9.0.0.0
3.1.0.0, 3.1.0.0 FE-1, 3.1.0.0 HF-1, 3.1.0.0 HF-2, 3.1.0.0 SE-1, 3.1.0.0 FE-2, 3.1.0.0 HF-3, 3.1.0.0 SE-2, 3.1.0.0 HF-4, 3.1.0.0 SE-3, 3.1.0.0 FE-3, 3.1.0.0 SE-4, 3.1.0.0 HF-5	9.1.0.0
3.1.0.1, 3.1.0.1 SE-1, 3.1.0.1 HF-1	9.1.0.1
3.1.0.2, 3.1.0.2 HF-1, 3.1.0.2 FE-1, 3.1.0.2 HF-2, 3.1.0.2 HF-3, 3.1.0.2 SE-1, 3.1.0.2 HF-4	9.1.0.2
3.1.0.3, 3.1.0.3 HF-1	9.1.0.3
3.1.0.4, 3.1.0.4 HF-2	9.1.0.4
3.1.0.5	9.1.0.5

Chapter 10

DSG Web UI

10.1 Introducing the Cloud Gateway Menu

This section provides an overview of available UI menus and sub menus.

10.1 Introducing the Cloud Gateway Menu

The Data Security Gateway (DSG) Web UI are a collection of DSG-specific UI screens under Cloud Gateway menu that are part of the ESA Web UI. The Cloud Gateway menu is enabled after the ESA patch for Cloud Gateway is installed in ESA.

The DSG menu includes UI screens that let you monitor clusters or nodes, view logs, create profiles and rules, create tunnels, create or upload certificates, and so on.

The ESA dashboard is as seen in the following figure.

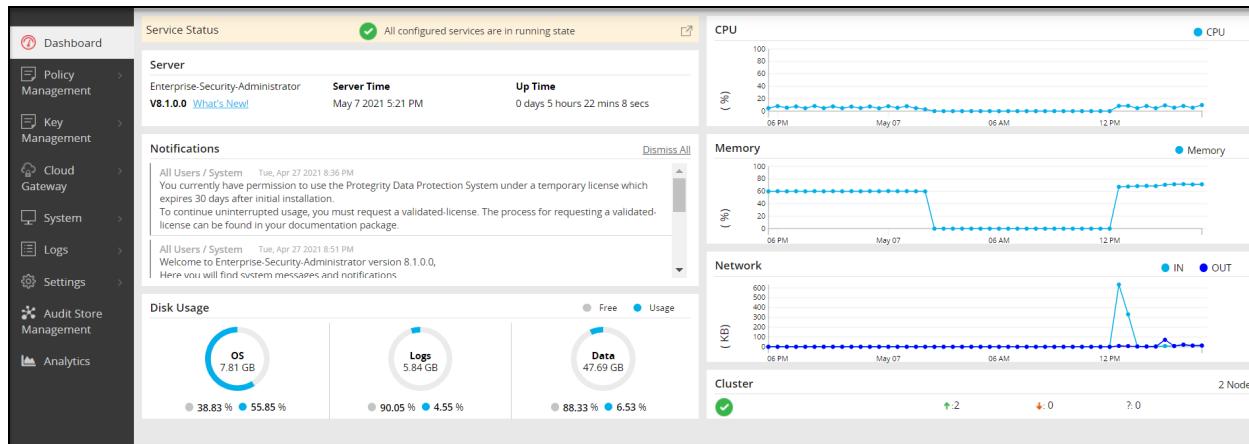


Figure 10-1: ESA Dashboard

The following table describes the Cloud Gateway sub menus as seen on the ESA Web UI:

Table 10-1: Cloud Gateway Menu

Menu	Sub menus	Description
Cluster	Monitoring	Displays all the nodes in a cluster. You can also add, delete, start, stop, and apply patches to node(s) in the cluster.
	Log Viewer	Displays consolidated logs across all DSG nodes.

Menu	Sub menus	Description
RuleSet	RuleSet	Displays the hierarchical set of rules. The rules are defined for each profile that is created under a service.
	Learn Mode	Displays hierarchical processing for a rule that is affected due to a triggered transaction request or response.
Transport	Certificates	Displays the certificates generated by or uploaded to the DSG.
	Tunnels	Displays the list of tunnels configured for the DSG. Tunnels are created specific to the protocol used.
Global Settings	Debug	The Debug tab lets you configure log settings, Learn mode settings, and set configurations that enable administrative queries.
	Global Protocol Stack	Apart from the settings that you configure for each service type, some settings affect all services that relate to a protocol type.
	Web UI	The Web UI tab lets you configure additional settings that impact how the UI is displayed.
Test Utilities*		The test utilities provide an interface where you can select the data security operation you want to perform, along with the DSG node, data elements available in the policies deployed at that node, and an external IV value for added security layer.

* - The Test Utilities menu appears only if the user logged in is granted the policy user permission.

Caution: It is recommended to create the Tunnel and Ruleset configurations on the ESA, such that the same configuration can be pushed simultaneously to all the ESA and DSG nodes in the cluster. If you do not create Tunnel and Ruleset configurations on the ESA, then any specific configuration created on the DSG node can be overridden by the configurations created on the ESA.

10.1.1 Cluster Menu

The Cluster menu includes the Monitoring and the Log Viewer tabs. Use the Cluster menu options to monitor the cluster health and view the cluster DSG logs.

10.1.1.1 Monitoring tab

The Monitoring tab displays information about the available nodes in a DSG cluster. You can perform the following functions from this screen:

Note: For more information about the sub-clustering feature, refer to the section [Overview of Sub-Clustering](#) and [Sub-Clustering FAQs](#).

- Add more nodes to the existing cluster based on your requirements
- Deploy the configurations to all the nodes in cluster
- Deploy to node groups
- Change groups on entire cluster
- Change groups on selected nodes
- Refresh all nodes together



- Start, stop, or restart individual or all nodes

Caution: Ensure that no configuration changes, such as editing rulesets, are made while applying the patch to the DSG nodes sequentially.

The Cluster Monitoring tab is as seen in the following figure.

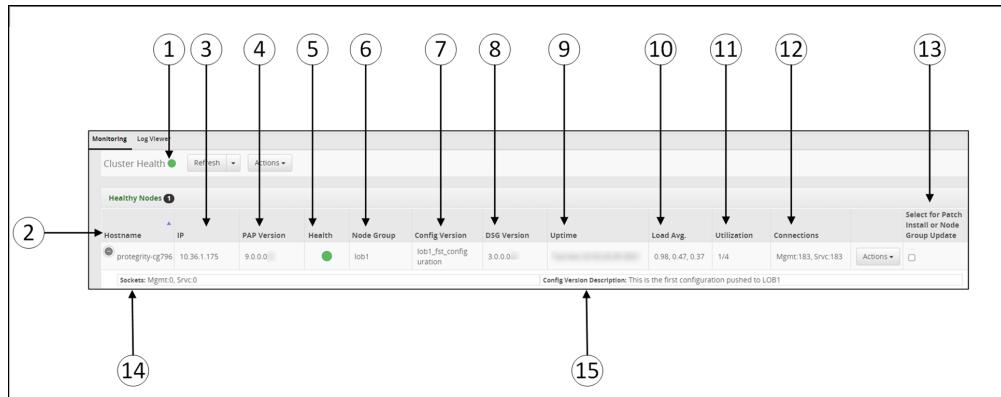


Figure 10-2: Cluster Monitoring screen

The following table provides the description for each column available on the UI.

Table 10-2: Cluster Menu Columns

Callout	Column	Description
1	Cluster Health*	Cluster health indicator in Green, Orange, or Red.
2	Hostname	Hostname of the DSG node.
3	IP	IP address of the DSG node.
4	PAP Version	Build version of the DSG Appliance.
5	Health	Status of the DSG node.
6	Node Group	The DSG nodes can be assigned to a node group to create sub-clusters. The node group name must contain alphanumeric characters or underscores. If the node group name is not specified, then by default the <i>default</i> node group name is assigned to a DSG node. Note: For more information about the sub-clustering feature, refer to the section <i>Overview of Sub-Clustering</i> and <i>Sub-Clustering FAQs</i> .
7	Config Version	The name of configuration version that is to be deployed to a particular node group. The tag name provided while deploying the configuration to a particular node group is displayed in the <i>Config Version</i> column in the Cluster tab. Note: For more information about the tag name, refer to this step .
8	DSG Version	Build version of the DSG.

Callout	Column	Description
9	Uptime	Time that the DSG has been running.
10	Load Avg	Average load on a process in the last five, ten, and fifteen minutes.
11	Utilization	Number of DSG processes vs. CPU cores
12	Connections	Total number of active connections for the node.
13	Select for Patch Installation or Node Group Update	Select for performing any node group update on a node. Caution: In this release, there is no patch available for the DSG appliance, so this option cannot be used to install a patch.
14	Socket	Total number of open sockets for the node.
15	Config Version Description	The additional information about the configuration version. Note: If the description is not provided while deploying the configurations to a particular node group, then this field will be empty. Note: For more information about the configuration version description, refer to this step .

Cluster and node health status color indication reference:

●	Node is healthy and services are running
●	Warning. Some services related to a node(s) need attention
●	Not running (or unreachable)

The following figure illustrates the actions for the *Cluster Monitoring* screen.

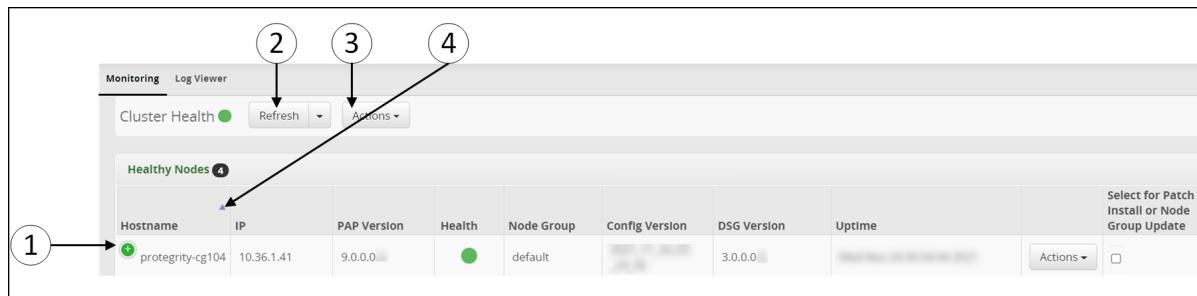


Figure 10-3: Action items in Cluster Monitoring

The following table provides the description for each action available on the Web UI.

Callout	Button	Description
1	Expand	Expand to view the other columns.
2	Refresh**	Refresh all the nodes in the cluster.
3	Actions***	Apply patch on cluster or selected nodes, change group on entire cluster or on selected nodes, or add a node to the cluster.
4	Order	
		The hostname or the IP address of the DSG nodes are sorted in the ascending order.
		The hostname or the IP address of the DSG nodes are sorted in the descending order.
5	Actions****	Start, stop, restart, apply a patch, change group, or delete a node.

**The Refresh drop down list provides the following options:

- Refresh: Refreshes details related to all nodes
- Start: Starts all nodes
- Stop: Stops all nodes
- Restart: Restarts all nodes

Note: The restart operation will not export the configurations, it will only restart all the nodes.

Note: The cluster start, stop, and restart operations will only work from the ESA.

- Deploy: Deploys the configuration changes on all the DSG nodes in the cluster

For more information about deploying the configurations to entire cluster, refer to the section [Deploying the Configurations to Entire Cluster](#).

- Deploy to Node Groups: Deploy the configurations to the selected node groups in the cluster

For more information about deploying the configurations to node groups, refer to the section [Deploying the Configurations to Node Groups](#).

Note: The Deploy and Deploy to Node Groups will push the configurations and restart the nodes.

***The Actions drop down list at the cluster level provides the following options:

- Apply Patch on Cluster: Applies a patch to all nodes in a cluster including the ESA.
- Apply Patch on Selected Nodes: Apply the same patch simultaneously on all selected nodes.
- Change Groups on Entire Cluster: Change the node group of all the DSG nodes in the cluster
- Change Groups on Selected nodes: Select the node and change the node group on that particular DSG node in the cluster
- Add Node: Add a DSG node to the cluster

For more information about adding a node to the cluster, refer to the section [Adding a Node to the Cluster](#).

Caution: Only 50 DSG nodes are tested as part of a single cluster that is associated with an ESA.

****The Actions drop down list at the individual node level provides the following options:

- Start: Start a node
- Stop: Stop a node
- Restart: Restart a node

Note: The restart operation will not export the configurations, it will only restart all the nodes.

- Apply Patch: Applies a patch to a single node

Note: Before applying a patch on a single DSG node, ensure that the same patch is applied on the ESA.

Caution: In this release, there is no patch available for the DSG appliance, so **Apply Patch** option cannot be used to install a patch.

- Change Groups: Changes the node group on an individual DSG node
- Delete: Delete a node

10.1.1.1 Deploying the Configurations to Entire Cluster

The configurations can be pushed to all the DSG nodes in the cluster. This action can be performed by clicking the **Deploy** option on the **Cluster** page or from the **Ruleset** page.

► To deploy the configurations to entire cluster:

1. In the ESA Web UI, navigate to **Cloud Gateway > Cluster**.
2. Select the **Refresh** drop down menu and click **Deploy**.

The following pop-up message occur on the **Cluster** screen.

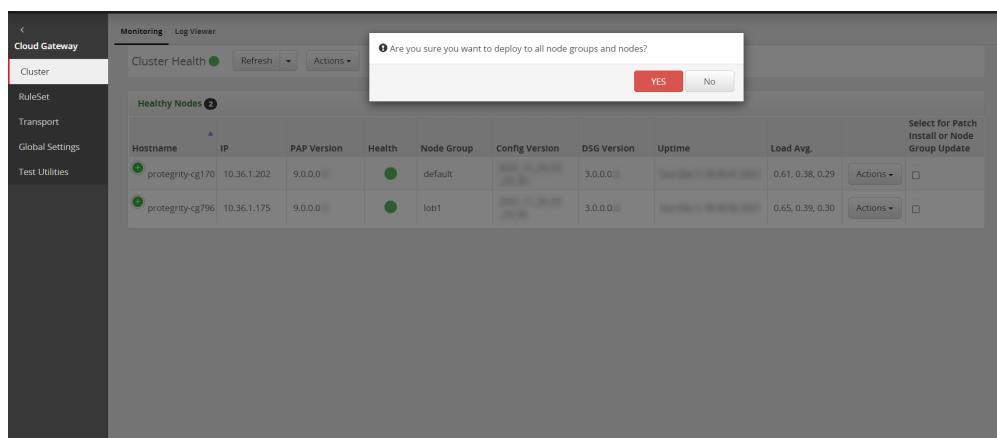


Figure 10-4: Cluster screen on deploying configurations to entire cluster

3. Click **YES** to push the configurations to all the node groups and nodes.
The configurations will be deployed to all the nodes in the entire cluster.

10.1.1.1.2 Deploying the Configurations to Node Groups

The configurations can be pushed to the selected node groups. The configuration will only be pushed to the DSG nodes associated with the node groups. This action can be performed by clicking the **Deploy to Node Groups** option on the **cluster** page or from the **Ruleset** page.

► To deploy the configurations to the selected node groups:

1. In the ESA Web UI, navigate to **Cloud Gateway > Cluster**.
2. Select the **Refresh** drop down menu and click **Deploy to Node Groups**.
3. Perform the following steps to deploy the configurations to the node groups.
 - a. Click **Deploy to Node Groups**.

The *Select node groups for deploy* screen appears.

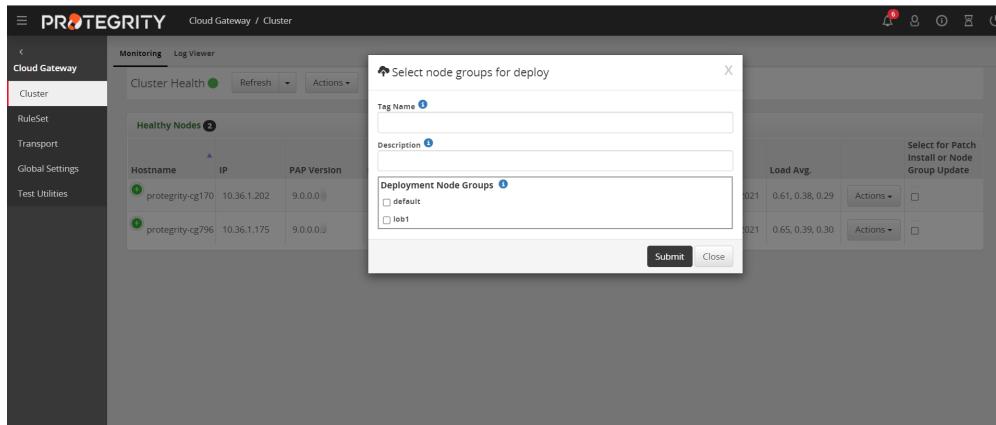


Figure 10-5: Deploy to Node Group screen

Note: The *default* and *lob1* are the node groups associated with the DSG nodes. When you add a node to cluster, a node group is assigned to that node.

For more information about adding a node and node group to the cluster, refer to the section [Adding a Node to the Cluster](#).

- b. Enter the name for the configuration version in the **Tag Name** field.

Note: The tag name is the version name of a configuration that is deployed to a particular node group. The tag name must be alphanumeric separated by spaces or underscores. If the tag name is not provided, then it will automatically generate the name in the *YYYY_mm_dd_HH_MM_SS* format.

- c. Enter the description for the configuration in the **Description** field.

Note: The user can provide additional information about the configuration that is to be deployed.

- d. On the **Deployment Node Groups** option select the node group to which the configurations must be deployed.
- e. Click **Submit**.

10.1.1.2 Log Viewer

The *Log Viewer* screen provides a unified view of the log messages across all the nodes in the cluster. This screen contains informational messages, warnings, and error messages that are generated across the nodes. The following table provides an overview of the type of log levels available for the DSG.

Table 10-3: Log types

Log Level	Description	Color
Debug	Debugging trace	N/A
Verbose	Additional information that can help a user with detailed troubleshooting	N/A
Information	Log entry for information purposes	N/A
Warning	Non-critical problem	Orange
Error	A problem that requires user's attention	Red

The Log Viewer tab is shown in the following figure.

The following figure illustrates the actions for the *Log Viewer* screen.

Figure 10-6: Log Viewer Menu

The following table provides the description for each column:

Table 10-4: Log Viewer Menu Columns

Callout	Column/Textbox/Button	Description
1	Host	Host name or IP address of the DSG node where the log message was generated.
2	PID	Captures the process identifier of the DSG daemons that generated the log message.
3	Timestamp (UTC)	Time recorded when an event for the log was generated. The time recorded is displayed in the Coordinated Universal Time (UTC) format.
4	Level	Severity level of the log message.
5	Module	Part of the program that generated the log.
6	Source	Procedure in the module that generated the log.



Callout	Column/Textbox/Button	Description
7	Message	A textual description of the event logged.

The following figure illustrates the actions for the *Log Viewer* screen.

Host	PID	Timestamp (UTC)	Level	Module	Source	Message
2.10.1.105	48183	2016-07-31T07:26:48.779019	Verbose	CommandRequestHandler	_handleRequest	Responding with 200 , body length: 375
2.10.1.105	48183	2016-07-31T07:26:48.491971	Verbose	CommandRequestHandler	_handleRequest	Processing request from 2.10.1.100 POST /
2.10.1.101	16102	2016-07-31T07:25:35.606104	Verbose	CommandRequestHandler	_handleRequest	Responding with 200 , body length: 375
2.10.1.101	16102	2016-07-31T07:25:35.203947	Verbose	CommandRequestHandler	_handleRequest	Processing request from 2.10.1.100 POST /
2.10.1.105	48183	2016-07-28T07:58:26.144699	Verbose	CommandRequestHandler	_handleRequest	Responding with 200 , body length: 375
2.10.1.105	48183	2016-07-28T07:58:25.956047	Verbose	CommandRequestHandler	_handleRequest	Processing request from 2.10.1.100 POST /
2.10.1.101	16102	2016-07-28T07:56:54.732935	Verbose	CommandRequestHandler	_handleRequest	Responding with 200 , body length: 376
2.10.1.101	16102	2016-07-28T07:56:54.387632	Verbose	CommandRequestHandler	_handleRequest	Processing request from 2.10.1.100 POST /
2.10.1.105	48183	2016-07-28T07:55:34.184811	Verbose	CommandRequestHandler	_handleRequest	Responding with 200 , body length: 375
2.10.1.105	48183	2016-07-28T07:55:33.926761	Verbose	CommandRequestHandler	_handleRequest	Processing request from 2.10.1.100 POST /
2.10.1.101	16102	2016-07-28T07:54:03.404987	Verbose	CommandRequestHandler	_handleRequest	Responding with 200 , body length: 376
2.10.1.101	16102	2016-07-28T07:54:03.015130	Verbose	CommandRequestHandler	_handleRequest	Processing request from 2.10.1.100 POST /
2.10.1.105	48183	2016-07-27T11:31:33.229957	Verbose	CommandRequestHandler	_handleRequest	Responding with 200 , body length: 375
2.10.1.105	48183	2016-07-27T11:31:33.040761	Verbose	CommandRequestHandler	_handleRequest	Processing request from 2.10.1.100 POST /
2.10.1.105	48183	2016-07-27T11:31:30.160177	Verbose	CommandRequestHandler	_handleRequest	Responding with 200 , body length: 375
2.10.1.105	48183	2016-07-27T11:31:29.875911	Verbose	CommandRequestHandler	_handleRequest	Processing request from 2.10.1.100 POST /
2.10.1.105	48183	2016-07-27T11:31:19.425444	Verbose	CommandRequestHandler	_handleRequest	Responding with 200 , body length: 375
2.10.1.105	48183	2016-07-27T11:31:19.255246	Verbose	CommandRequestHandler	_handleRequest	Processing request from 2.10.1.100 POST /
2.10.1.101	16102	2016-07-27T11:29:57.477260	Verbose	CommandRequestHandler	_handleRequest	Responding with 200 , body length: 376
2.10.1.101	16102	2016-07-27T11:29:57.167576	Verbose	CommandRequestHandler	_handleRequest	Processing request from 2.10.1.100 POST /

Figure 10-7: Log Viewer Menu

The following table provides the description for each action available on the UI.

Table 10-5: Log Viewer Menu Action items

Callout	Column/Textbox/Button	Description
1	Search Log	Search the entire log archive that is collectively maintained across all the DSG nodes within the cluster.
2	Clear	Clear logs that appear on the <i>Log Viewer</i> screen.
3	Refresh	Refresh the <i>Log Viewer</i> screen to show the latest records.

Note: To view the message related to an entry, select the log entry.

10.1.1.2.1 Clearing Records from the Log Viewer Screen

You can use the clear functionality to clean up the logs and view the newly generated logs on the *Log Viewer* screen.

Clearing the *Log Viewer* screen removes the entries that are currently displayed. You can view all the archived logs even after the records are cleared.

Note: The logs are cleared only from the *Log Viewer* screen. The logs are not cleared or deleted from the appliance and are available for future reference. To access these logs, on the DSG node, click **Logs > Appliance**. Under **Cloud Gateway - Event Logs**, select **gateway**.

10.1.1.2.1.1 Retrieving Archived Logs

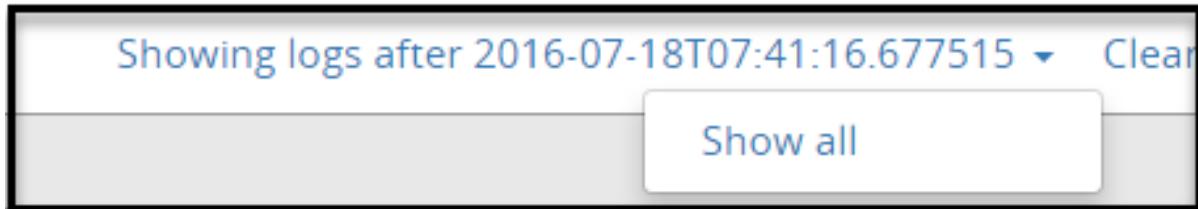
You can use the clear functionality to clean up the logs and view the newly generated logs in the Log Viewer screen. Clearing the Log Viewer screen removes the entries that are currently displayed. You can view all the archived logs even after the records are cleared.

► To Retrieve Archived Logs after Clearing the *Log Viewer* Screen:

1. Click **Clear** to remove all the records from the *Log Viewer* screen.

Note: The logs are cleared only from the *Log Viewer* screen. The logs are not cleared or deleted from the server and are available for future reference.

2. Click Refresh ().
A link displaying the timestamp of the last updated record appears.



3. Click **Showing logs after <timestamp> > Show all**.
The latest 1000 logs appear in the *Log Viewer* screen.

10.1.1.2.2 Fetching Log Records

In the *Log Viewer* screen, you have the option to view the new log records using either Manual refresh or Auto refresh.

10.1.1.2.2.1 Manually Refreshing Records

This section describes the steps to refresh records manually.

► To Manually Refresh records in the *Log Viewer* screen:

1. On the ESA Web UI, navigate to **Cloud Gateway > Cluster > Log Viewer**.
2. Click **Refresh** ().
You can view the top 1000 logs generated by different nodes in the cluster.

10.1.1.2.2.2 Automatically Refreshing Records

This section describes the steps to refresh records automatically.

► To Automatically Refresh the *Log Viewer* screen:

1. In the ESA Web UI, navigate to **Cloud Gateway > Cluster > Log Viewer**.
2. Click **Auto Refresh** ().

You can view the top 1000 logs generated by different nodes in the cluster.

3. Set the frequency of the screen update.

Note: By default, the **Auto Refresh** option is turned off.

10.1.1.2.3 Searching Log Records

You can also filter the specific log records from the *Log Viewer* screen based on the search criteria entered.

► To search for new records:

1. In the ESA Web UI, navigate to **Cloud Gateway > Cluster > Log Viewer**.
2. Type the text in the **Search** textbox and press **ENTER**.

The records based on the search criteria appear on the *Log Viewer* screen.

Note: The Search textbox accepts numerical values. For example, if you want to search for logs that occurred on 2016-07-18T07:41:15.429350 timestamp, you must remove the date and time separators, and then search for 20160718074115429350.

The search for logs is not limited to the records that appear on the screen. When a user clicks search, all the log records that are present on the screen as well as on the server are retrieved.

10.1.2 Ruleset Menu

The RuleSet menu includes the RuleSet and the Learn Mode tabs. Use the RuleSet menu to create services and monitor the rulesets using Learn Mode.

Note: For more information about the Web UI changes and sub-clustering, refer to the section [Overview of Sub-Clustering](#) and [Sub-Clustering FAQs](#).

10.1.2.1 RuleSet tab

The Ruleset tab provides you the capability to create a hierarchical rule pattern based on the service type. The changes made to the Ruleset tree require deployment of configuration to take effect.

Note: For more information about the sub-clustering feature, refer to the section [Overview of Sub-Clustering](#) and [Sub-Clustering FAQs](#).

The RuleSet tab is shown in the following figure:



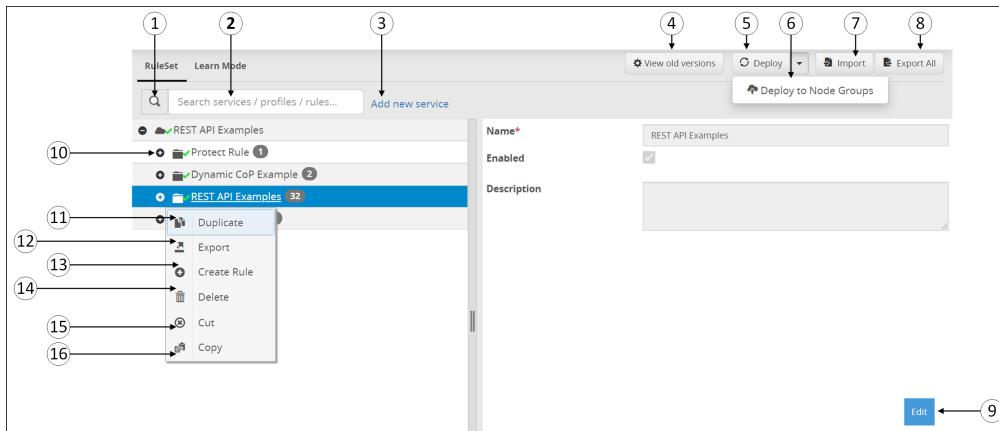


Figure 10-8: RuleSets Tree

The following table provides the description for each of the available RuleSet options:

Table 10-6: RuleSets Menu UI Columns

Callout	Icon	Column/Textbox/Button	Description
1		Search	Click to search for service, profile, or rules.
2		Search textbox	Provide service, profile, or rule name.
3		Add new service	<p>Add a new service-level based on the service type used.</p> <p>Note: You can create only one service for every service type. For more information about creating services, refer to the section Best Practices.</p>
4		View Old Versions	<p>Click to view archived Ruleset configuration backups.</p> <p>For more information about the older version of ruleset configurations, refer to the sections Understanding the Ruleset Versioning and Working with Ruleset Versions.</p>
5		Deploy	<p>Deploy the configurations to all the DSG nodes in the cluster. The Deploy operation will export the configurations and restart all the nodes.</p> <p>Note: Deploy can also be performed from the Cluster tab.</p> <p>Note: For more information about deploying the configurations to entire cluster, refer to the section Deployment.</p>

Callout	Icon	Column/Textbox/Button	Description
			<p><i>Deploying the Configurations to Entire Cluster.</i></p>
6		Deploy to Node Groups	<p>Deploy the configurations to the selected node groups in the cluster. The Deploy to Node Groups operation will export the configurations and restart the nodes associated with the node groups.</p> <p>Note: For more information about deploying the configurations to node groups, refer to the section <i>Deploying the Configurations to Node Groups</i>.</p>
7		Import	<p>Import the Ruleset tree to the Web UI.</p> <p>Note: You can only upload files with <i>.zip</i> extension. Ensure that the service exists as part of the Ruleset before you import a configuration exported at Profile level. Ensure that the directory structure that the exported <i>.zip</i> maintains is replicated when you repackage the files for import. Also, the JSON files must be valid.</p> <p>Caution: If an older ruleset configuration <i>.zip</i> created using any older DSG version, that includes a GPG ruleset with key passphrase defined, is imported, then the DSG does not encrypt the key passphrase.</p>
8		Export All	<p>Export the Ruleset tree configuration.</p> <p>Note: The rules are downloaded in a <i>.zip</i> format.</p>
9		Edit	Edit the service, profile, or rule details as per requirement.

Callout	Icon	Column/Textbox/Button	Description
10		Expand Rule	Expand the rule tree and view child rules.

If you want to further work with rules, right-click any rule to view a set of sub menus. The sub menu options are seen in above figure. The options are described in the following table.

Table 10-7: RuleSets Sub Menu UI Columns

Callout	Icon	Column/Textbox/Button	Description
11		Duplicate	Duplicate a service, profile, or rule to create a copy of these Ruleset elements. Note: You must refresh the ruleset page after a service, profile, or rule is duplicated.
12		Export	Export the Ruleset tree configuration at Service or Profile level. All the child rules under the parent Service or Profile are exported. Note: The rules are downloaded in a .zip format.
13		Create Rule	Add child rule under the parent rule.
14		Delete	Delete the selected rule.
15		Cut	Cut the selected rule from the parent rule.
16		Copy	Copy the selected rule under a parent.
17		View Configuration	View the configuration of the rule in the JSON format. You can copy the JSON format of the rule and pass it as parameter value in the header of the Dynamic CoP ruleset. Note: This option is available only for the individual rules.

Instead of cut and copy a rule to change its hierarchy among siblings, you can also drag a sibling rule and change its positioning.

When the drop is successful, a green tick icon () is displayed as shown in the following figure.

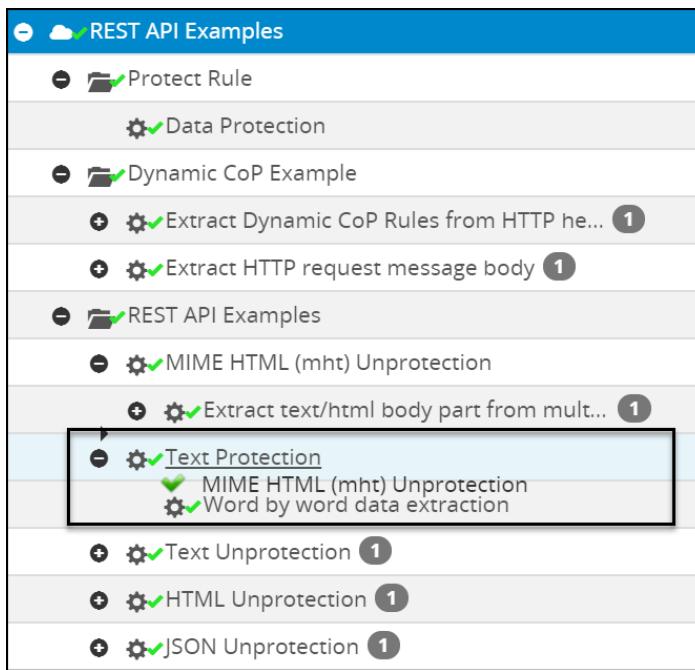


Figure 10-9: Drag and Drop Sibling - Correct Hierarchy

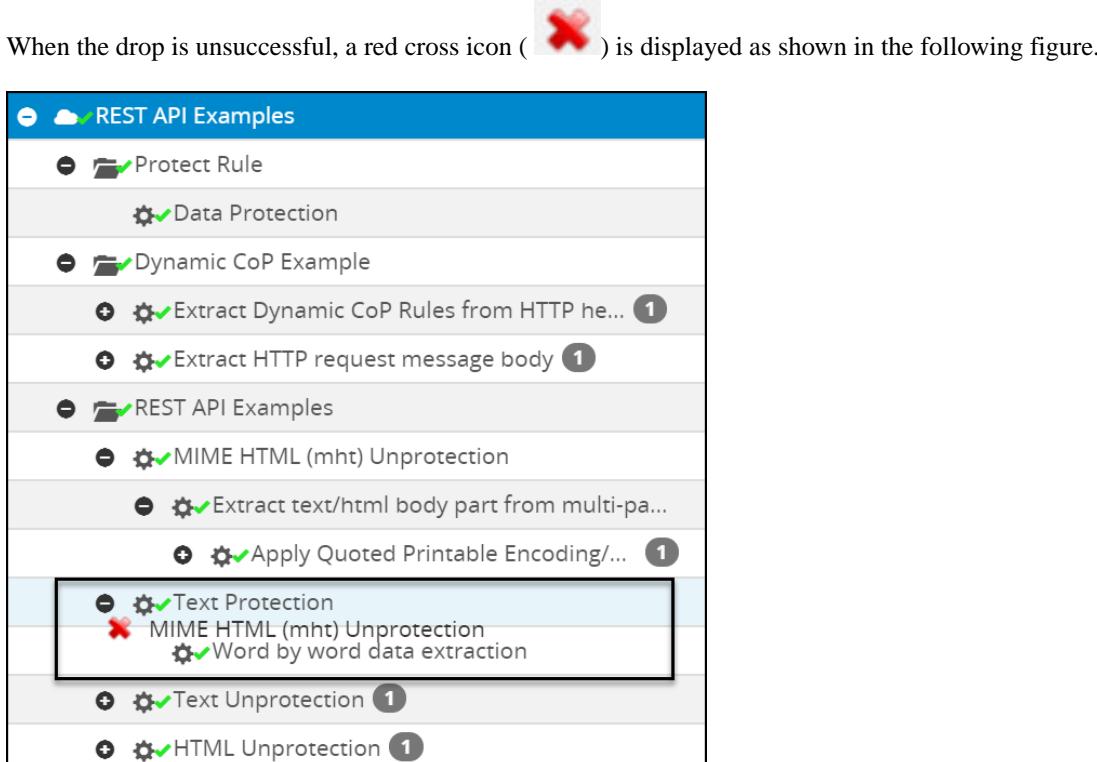


Figure 10-10: Drag and Drop Sibling - Incorrect Hierarchy

Note: A log is generated in the Forensics screen every time you cut, copy, delete, or reorder a rule from the Ruleset screen in the ESA.

The following figure shows a service with Warning indication.

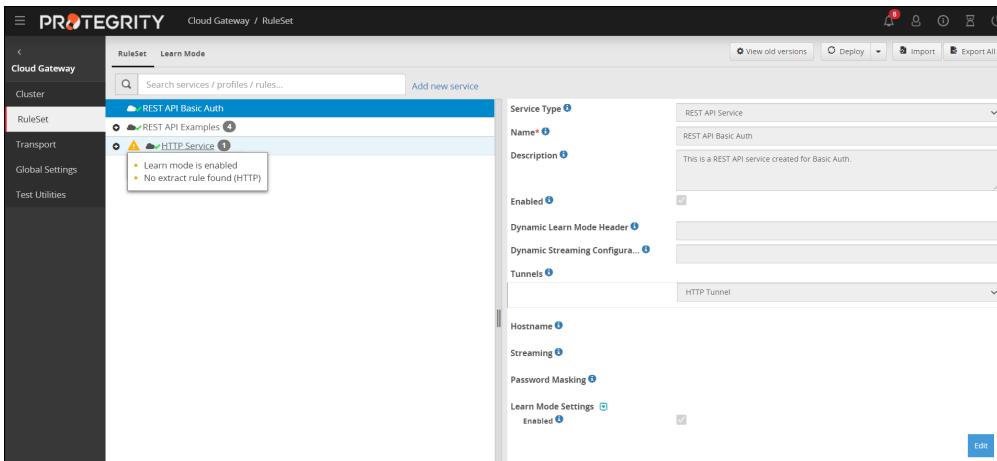


Figure 10-11: Ruleset with Warning symbol

Table 10-8: RuleSets with Warning symbol

Callout	Icon	Column/Textbox/Button	Description
1	⚠	Warning	The Warning symbol is seen on the service when the child rule is not created or when <i>Learn Mode</i> is enabled.

10.1.2.1.1 Deploying the Configurations to Entire Cluster

The configurations can be pushed to all the DSG nodes in the cluster. This action can be performed by clicking the **Deploy** button on the **RuleSet** tab or from the **Cluster** tab.

► To deploy the configurations to entire cluster:

1. In the ESA Web UI, navigate to **Cloud Gateway > Ruleset**.
2. Click **Deploy**.

The following pop-up message occur on the **Ruleset** screen.

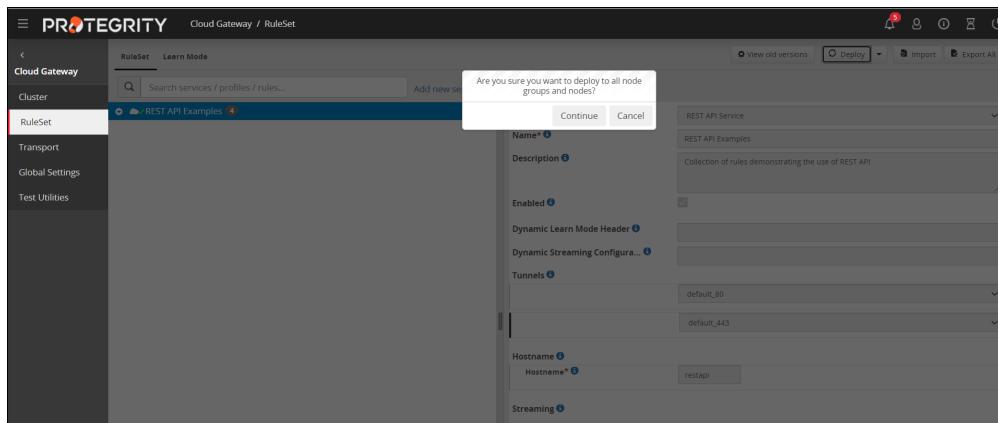


Figure 10-12: Ruleset screen on deploying configurations to entire cluster

3. Click **Continue** to push the configurations to all the node groups and nodes.
The configurations will be deployed on the entire cluster.

10.1.2.1.2 Deploying the Configurations to Node Groups

The configurations can be pushed to the selected node groups. The configuration will only be pushed to the DSG nodes associated with the node groups. This action can be performed by clicking the **Deploy to Node Groups** button on the **RuleSet** page or from the **Cluster** tab.

► To deploy the configurations to the selected node groups:

1. In the ESA Web UI, navigate to **Cloud Gateway > RuleSet**.
2. Perform the following steps to deploy the configurations to the node groups.
 - a. Click **Deploy to Node Groups**.

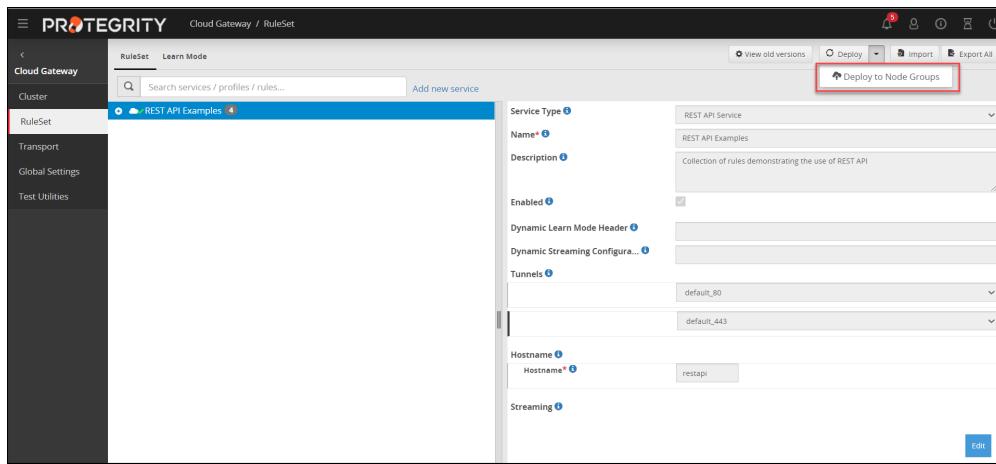


Figure 10-13: Deploy to Node Groups option on the Ruleset screen

The *Select node groups for deploy* screen appears.

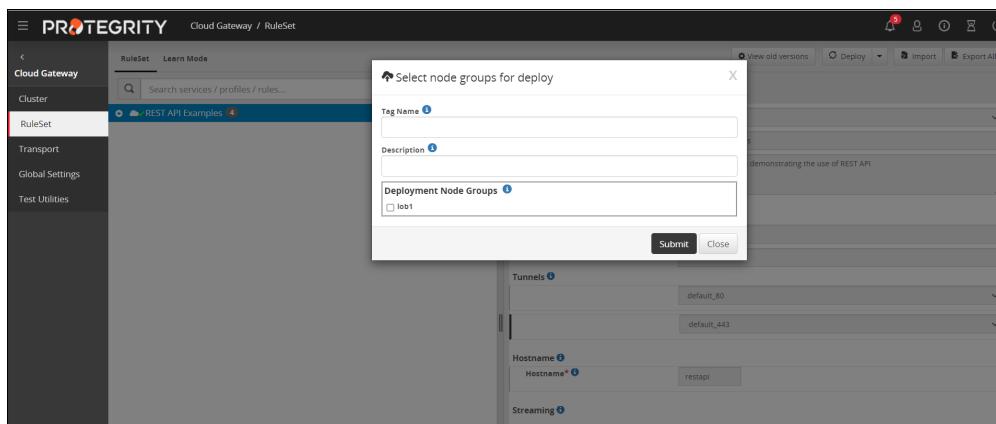


Figure 10-14: Deploy to Node Group screen

- b. Enter the name for the configuration version in the **Tag Name** field.

Note: The tag name is the version name of a configuration that is deployed to a particular node group. The tag name must be alphanumeric separated by spaces or underscores. If the tag name is not provided, then it will automatically generate the name in the *YYYY_mm_dd_HH_MM_SS* format.

- c. Enter the description for the configuration in the **Description** field.

Note: The user can provide additional information about the configuration that is to be deployed.

- d. On the **Deployment Node Groups** option select the node group to which the configurations must be deployed.
e. Click **Submit**.

10.1.2.1.3 Understanding the Ruleset Versioning

This section provides information about the different types of version visible on the ruleset page on clicking the **View Older Versions** button. After deploying a configuration to particular node group or to entire cluster, a backup of these configurations are saved in **View Older Versions** on the **Ruleset** page. The most recent deployed configuration for a particular node group is shown as **Deployed** status on viewing the older versions. There are tagged and untagged versions seen on viewing the older versions. You can create a tagged or untagged version by using these [steps](#).

The following figure shows the *Ruleset versioning* screen.

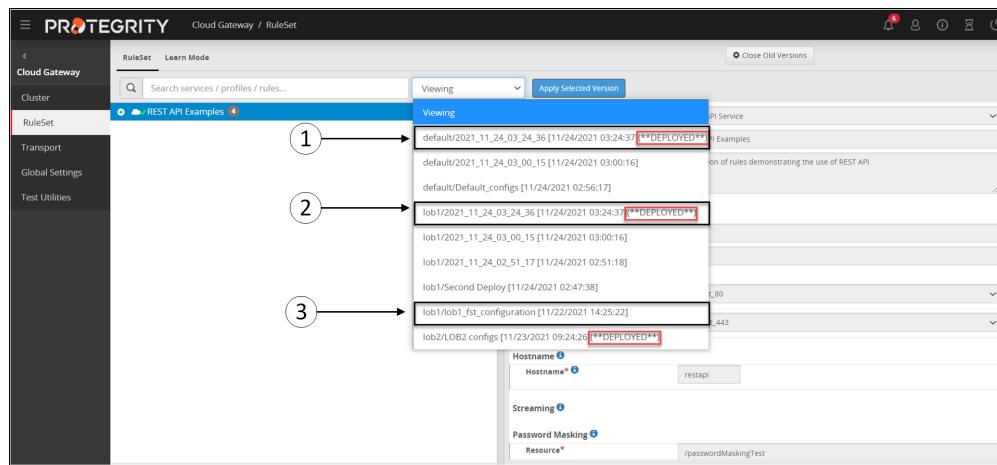


Figure 10-15: Ruleset Versioning

The following table provides the description for the deployed configurations.

Table 10-9: Ruleset Versions

Callout	Description
1	<p>The configuration is deployed to <i>default</i> node group and you can see Deployed status for this configuration version. This is the most recent deployed configuration version for the <i>default</i> node group with Deployed status.</p> <p>Note: Each node group will have Deployed status for the most recent configuration version.</p>
2	<p>The configuration is deployed to <i>lob1</i> node group and the configuration version is untagged. As the version is untagged, it will automatically generate the name with timestamp in the <i>YYYY_mm_dd_HH_MM_SS</i> format.</p> <p>Note: Each node group will archive the three most recent untagged version.</p> <p>For more information about changing the default values for untagged version, refer to the note Configuring the default value.</p>

Callout	Description
3	<p>The configuration is deployed to <i>lob1</i> node group and the configuration version is tagged. While deploying the configuration to default node group the <i>lob1_fst_configuration</i> tag name was provided to configuration version.</p> <p>Note: Each node group will archive the ten most recent tagged version. For more information about changing the default values for tagged version, refer to the note Configuring the default value.</p>

10.1.2.1.4 Working with Ruleset Versions

Each time a configuration is changed and deployed, the DSG creates a backup configuration version. You can apply an earlier configuration version and make it active, in case you want to revert to the older configuration version.

► To view older configuration versions:

1. On the DSG Web UI, navigate to **Cloud Gateway > Ruleset**.

The following figure shows the *Ruleset versioning* screen.

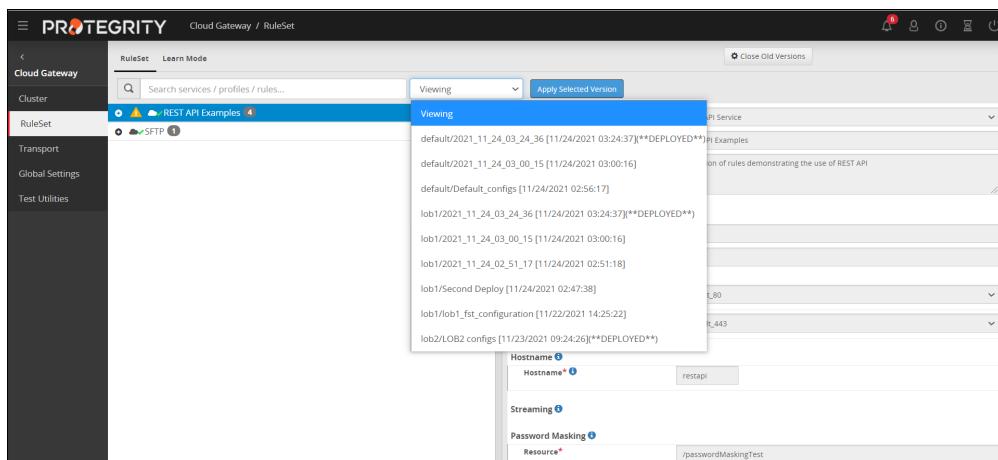


Figure 10-16: Ruleset Versioning

Note: For more information about the configuration versions, refer to the section [Understanding the Ruleset Versioning](#).

Note: If you want to change the number of tagged or untagged versions that a node can store, then on the DSG node, login to the OS console. Navigate to the `/opt/protegility/alliance/config/webinterface` directory. Edit the following parameter in the `nodeGroupsConfig.json` file.

`no_of_node_group_deployed_archives = <number_of_untagged_versions_to_be_stored>`

The default value for the untagged version is set at 3.

`no_of_node_group_deployed_tag_archives = <number_of_tagged_versions_to_be_stored>`

The default value for the tagged version is set at **10**.

2. Click **View Old Versions**.
3. Click the **Viewing** drop-down to view the available versions.
4. Select a version.
The left pane displays the Services, Profiles, and Rules that are part of the selected version.
5. Click **Apply Selected Version** to make the version active or click **Close Old Versions** to exit the screen.
6. Click **Deploy or Deploy to Node Groups** to save changes.

Note: For more information about deploying the configurations to entire cluster or the node groups, refer to the sections [Deploying the Configurations to Entire Cluster](#) and [Deploying the Configurations to Node Groups](#).

Note: It is recommended that any changes to the Ruleset configuration is made through the Cloud Gateway menu available on the ESA Web UI. Any changes made to the Ruleset configuration from the DSG Web UI of an individual node are overridden by the changes made to the ruleset configuration from the ESA Web UI. After overriding, the older Ruleset configuration on individual nodes is displayed as active and no backup for this configuration is maintained.

10.1.2.2 Learn Mode

Learn Mode tab provides a consolidated view of all message recorded by the DSG cluster. It allows you to consider messages exchanged through the DSG nodes and study the payloads as they are seen by the DSG. Understanding how messages are structured enables you to set the appropriate rules which will transform the relevant parts in it before it is forwarded on.

The Learn Mode tab is shown in the following figure.

The screenshot shows the Learn Mode interface with various numbered callouts pointing to specific elements:

- 1: Search bar.
- 2: Received (UTC) column header.
- 3: PID column header.
- 4: Source column header.
- 5: Destination column header.
- 6: Service column header.
- 7: Hostname column header.
- 8: Message column header.
- 9: Rule configuration sidebar.
- 10: Rule summary table.
- 11: Rule details table.
- 12: Rule payload table.
- 13: Rule payload table.
- 14: Rule payload table.

Received (UTC)	PID	Source	Destination	Service	Hostname	Message	Processing Time (ms)
2017-04-03 10:35:46.119855	54891	restapi	10.91.1.162	REST API Exampl...	CG1	200 OK (2017-04-03 10:35:21.905697 UTC)	0.691
2017-04-03 10:35:21.905697	54891	10.91.1.162	restapi	REST API Exampl...	CG1	POST https://restapi/protect/text	24187.850
2017-04-03 10:33:08.016933	24858	restapi	10.91.1.162	REST API Exampl...	CG1	200 OK (2017-04-03 10:33:08.014555 UTC)	0.211
2017-04-03 10:33:08.014555	24858	10.91.1.162	restapi	REST API Exampl...	CG1	GET https://restapi/protect/text	1.463

Below the table are buttons for "Show 50 entries" and navigation arrows. The rule configuration sidebar includes tabs for "Flow", "Host:CG1", "Tunnel:default_443", and "Service:REST API Examples".

Figure 10-17: Learn Mode Screen

The following table provides the description for each column available on the Web UI.

Table 10-10: Learn Mode Menu Columns

Callout	Column/Textbox/Button	Description
1	Received (UTC)	Time when the transaction is triggered. The time recorded is displayed in the Coordinated Universal Time (UTC) format.
2	PID	Process Identifier that has carried the request or response transaction on the gateway machine.
3	Source	Source IP address or hostname in the request.
4	Destination	Destination IP address or hostname in the request.
5	Service	Service name to which the transaction belongs.
6	Hostname	DSG node hostname where the request was received and processed.
7	Message	Provides information about the type of message.
8	Processing Time (ms)	Time required to complete the transaction.
9	Rules Filters	Filter the rules based on the selected option for a transaction.
10	Filter Summary	Summary of rule details, such as, Elapsed time, result, and Action Count
11	Message Difference	Difference between the message received by the rule and the message processed by the rule.
12	Wrap lines	Select to break the text to fit in the readable view.
13	View in Binary	<p>View message in hexadecimal format.</p> <p>Note: If you want to view a payload such as .zip, .pdf, or more, you can use the View in Binary option.</p>
14	Download Payload	Click to download large payloads that cannot be completely displayed on the screen.
**	Failed Transaction (in red color)	Any failed transaction is highlighted in the color red.

The following figure illustrates the actions in the *Learn Mode* screen.

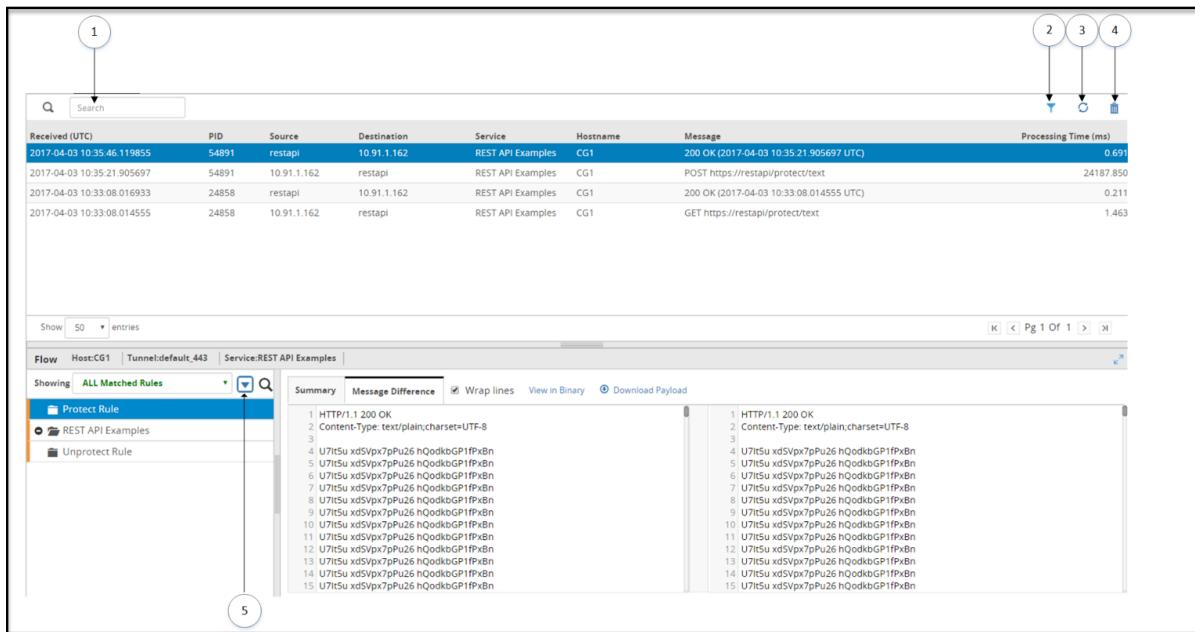


Figure 10-18: Action items in the Learn Mode screen

The following table provides the description for each action available on the Web UI.

Table 10-11: Learn Mode Action items

Callout	Column/Textbox/Button	Description
1	Search log	Search the learn mode content.
2	Column Filters	Apply column filters for each column to filter or search records based on the string and regex pattern match.
3	Refresh	Refresh the list.
4	Reset	Logs from the server are purged.
5	Collapse/Expand tree	Collapse or expand the rule tree.

You can select a record in the *Learn Mode* screen to view details regarding the matched and unmatched rules for that entry. If the size of the message exceeds the limit, then a message *Contents of the selected record are too large to be displayed* appears.

Note:

The Learn Mode logs that are generated over time can be scheduled for cleanup regularly. You can click **System > Task Scheduler**, select the **Learn Mode Log Cleanup** scheduled task, and then click **Edit** to modify the scheduled task that initiates the **learnmodecleanup.sh** file at regular intervals. The scheduled task can be set to *n* hours or days based on your preference. The default recommended frequency is **Daily-Every Midnight**.

In addition to setting the task, you can define the duration for which you want to archive the Learn Mode logs. The following image displays the **Learn Mode Log Cleanup** scheduled task.

6 - Learn Mode Log Cleanup - 0 0 * * *

Cluster task Is it a Cluster task? (Cluster task may be automatically updated among cluster nodes)

Name: Learn Mode Log Cleanup

Description: Removes old learn mode log files

Frequency

Daily - Every Midnight

Expression: 0 0

Minutes: 0

Days: Every Day

Days of the week: Every DOW

Description: 00:00 every day

Task next run: Tomorrow at 12:00 AM

Command Line

/opt/protegility/alliance/bin/scripts/learnmodecleanup.sh 10 DAYS

Figure 10-19: **Learn Mode Log Cleanup** scheduled task

The following table provides sample configurations:

Table 10-12: Sample configurations for Learn Mode - Log retention

Frequency	Command line value	Retain the logs for	Default values
Daily-Every Midnight	/opt/protegility/alliance/bin/scripts/learnmodecleanup.sh 10 DAYS	Last 10 DAYS	Days can be set between 1 to 366
Every hour	/opt/protegility/alliance/bin/scripts/learnmodecleanup.sh 10 HOURS	Last 10 HOURS	Hours can be set between 1 to 23

Note: If a numeric value is set without the **HOURS** or **DAYS** qualifier, then **DAYS** is considered as the default.

For more information about setting a scheduled task, refer to the [Protegility Appliances Overview Guide 9.1.0.5](#).

10.1.3 Transport Menu

The Transport Menu allows configuration of the transport layer of communication. The Transport Menu includes the Certificates tab and the Tunnels tab.

10.1.3.1 Certificates/Key Material tab

The Certificates/Key Material tab lets you configure TLS/SSL certificates for SSL Termination by DSG. This tab displays key material and other files in three different subtabs.

The Certificates/Key Material tab and subtabs are shown in the following figure.



Figure 10-20: Certificates/Key Material tab

The following table describes the available subtabs:

Table 10-13: Certificates/Key Material Subtabs

Callout	Column/Textbox	Description
1	Certificates	View self-generated or trusted certificates.
2	Keys	View paired keys associated with certificates and unpaired keys.
3	Other Files	View other files such as GPG data, etc.
4	Upload	Upload a certificate, key, or other files.

10.1.3.1.1 Certificates Subtab

The Certificates subtab displays certificates that are available in DSG when it is installed. The certificates uploaded to DSG are also displayed in this subtab. Other important information such as paired key, validity, and last modified date is also displayed.

Note: A certificate and key that is paired displays a () icon indicating that the certificate is ready to use. A certificate or key without any pairing is indicated with a () icon. If a certificate or key has expired, it is indicated with a () icon. Files available in the Other Files subtab will always be marked with a () icon.

Note:

The **Cloud Gateway Certificate Expiration Check** scheduled task is created by default to alert about certificates that are due to expire in the next 30 days.

For more information about creating scheduled tasks, refer to section *Scheduled Appliance Tasks* in the *Protegility Appliances Overview Guide 9.1.0.5*.

Note:

Before you regenerate any default expired certificates, ensure that the best practices for certificates and keys are noted.

For more information about the best practices, refer to [Default Certificates and Keys](#).

The Certificates subtab is shown in the following figure.

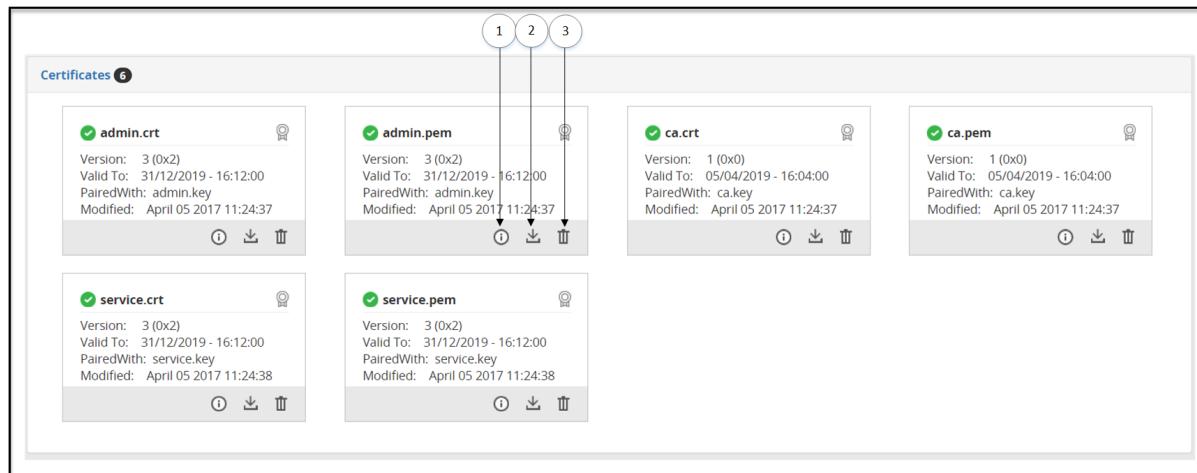


Figure 10-21: Certificates Subtab

The following table describes the available options:

Table 10-14: Certificates subtab options

Callout	Icon (if any)	Column/Textbox	Description
1		Information	View Certificate details.
2		Download	Download a certificate.
3		Delete	Delete a Certificate.

10.1.3.1.2 Keys Subtab

The Keys subtab displays the keys paired with the certificates and the keys that are no longer paired with a certificate. Keys cannot be downloaded, but you can either view key information or delete a key .

Note: A certificate and key that is paired displays a icon indicating that the certificate is ready to use. A certificate or key without any pairing is indicated with a icon. If a certificate or key has expired, then it is indicated with a icon. Files available in the **Other Files** subtab will always be marked with a icon.

Note:

The supported key formats that you can upload are **.crt**, **.csr**, **.key**, **.pgp**, **.pub**, and **.pem**. For any private key without an extension, when you click **Deploy to All Nodes**, the permissions for the key changes to **755** making it world readable. To restrict the permissions, ensure that you generate the key with the **.key** extension.

The keys uploaded to the DSG can either be a non-encrypted private key or an encrypted private key. For either of the key types uploaded, the DSG ensures that the keys in the DSG ecosystem are always present in an encrypted format. When a non-encrypted

private key is uploaded to the DSG, you are presented with an option to encrypt the key. If you choose to encrypt the key, DSG requests for a password for encrypting the key before it is stored on the DSG.

Note:

It is recommended that any non-encrypted private key is encrypted when uploaded to the DSG.

Note:

It is recommended that any key uploaded to the DSG is of RSA type and a minimum of 3072-bits for optimum security.

10.1.3.1.3 Others Files Subtab

The Other Files subtab displays files that were either uploaded to support GPG encryption-decryption, generated when DSG was installed, default files, and so on.

Note: A certificate and key that is paired displays a () icon indicating that the certificate is ready to use. A certificate or key without any pairing is indicated with a () icon. If a certificate or key has expired, it is indicated with a () icon. Files available in the Other Files subtab will always be marked with a () icon.

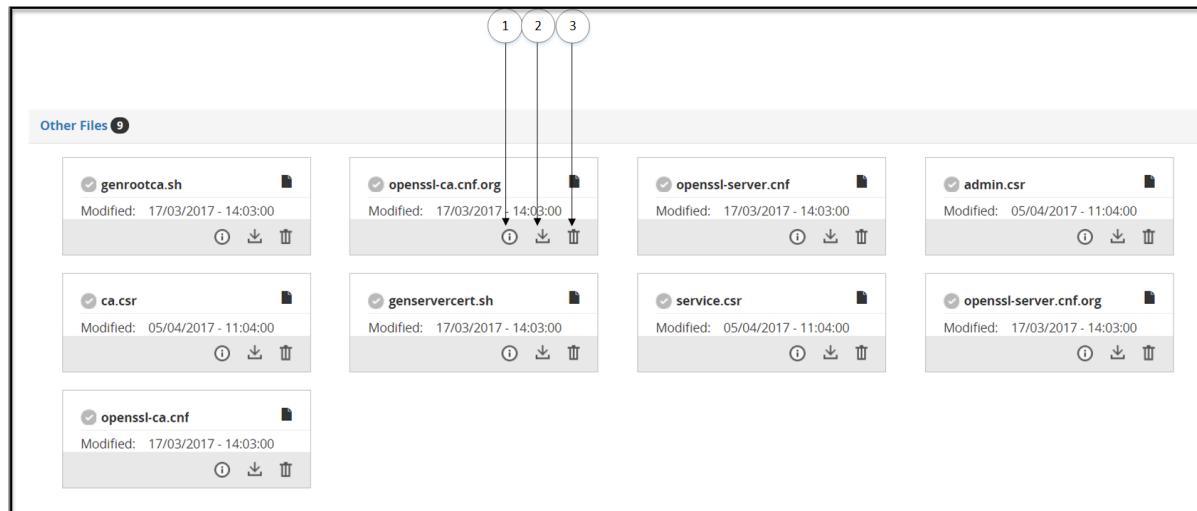


Figure 10-22: Other Files Subtab

The following table describes the available subtabs:

Table 10-15: Other Files Subtab Options

Callout	Icon (if any)	Column/Textbox	Description
1		Information	View Certificate details.
2		Download	Download a certificate.
3		Delete	Delete a Certificate.

10.1.3.1.4 Upload

You can upload predefined certificates and paired keys using the **Upload** option in the **Certificates** tab.

The *Upload Certificate/Key* screen is shown in the following figure:

Note: After clicking **Upload Certificate**, you can either upload a key or a certificate. When you upload a certificate, the password field does not appear.

After you click **Choose File** to select the key file, you must click **Upload Certificate**. Enter the password, and then click **Upload Certificate** again.

Note:

It is recommended that upload of any certificate or key is performed on the ESA. If the certificate is uploaded to a DSG node and configurations is deployed from ESA, then the changes made on the DSG node are overwritten by the configuration pushed by the ESA.

Note:

Ensure that the passphrase for any key that is uploaded to the DSG Web UI is of minimum 8 character length.

If the key you uploaded is an encrypted private key, then you must enter the password for the key.

If the key you uploaded is a non-encrypted private key, an option is presented to encrypt the private key. If you select the option, you must provide a password that the DSG uses to encrypt the non-encrypted private key before it is stored internally.

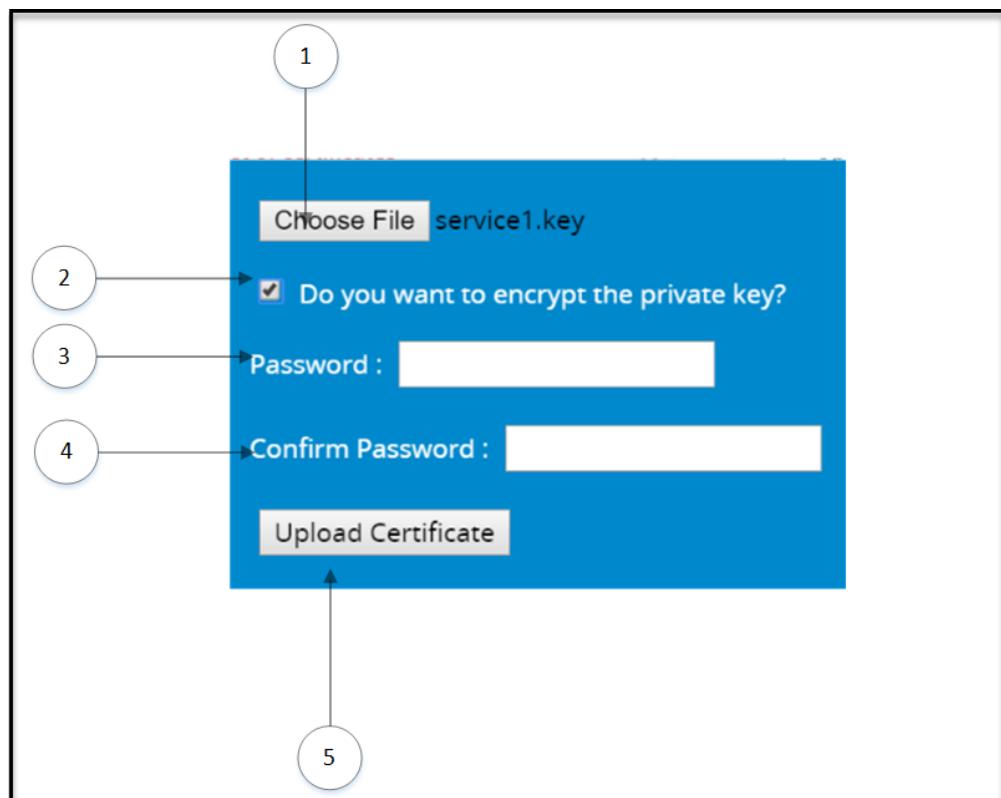


Figure 10-23: Upload Key/Certificate screen

The following table describes the available options:

Table 10-16: Upload screen UI options

Callout	Column/Textbox/Button	Description
1	Choose File	Select certificate and key files to upload. Note: You cannot upload multiple files in an instance. You must first upload the certificate file, and then the paired .key file. If you upload unpaired keys or certificates, then they are not displayed on the <i>Certificate</i> screen.
2*	Do you want to encrypt the private key	Select the check box to encrypt a non-encrypted private key. If you clear the check box, then the private key will be uploaded without encryption. Note: It is recommended that any non-encrypted private is encrypted when uploaded to the DSG.
3*	Password	Enter the password for an encrypted private key. For a non-encrypted private key, provide a password that will be used to encrypt the key. Caution: The DSG supports ASCII passwords for keys. If your private key is encrypted with any other character password, then ensure that it is changed to an ASCII password. For more information about password policies, refer to section <i>4.4.3.2 Strengthening Password Policy</i> in the <i>Protegility Appliances Overview Guide 9.1.0.5</i> .
4*	Confirm Password	Re-enter the password
5	Upload Certificate	Upload the certificate or .key file. Note: If you upload a private key without an extension, then ensure that you append the .key extension to the key.

*-Appears only when a key is uploaded.

10.1.3.1.5 Delete

You can delete existing certificates, keys, and other files using the Delete option in the Certificates/Key Material tab.

The *Delete Certificate* screen is shown in the following figure:



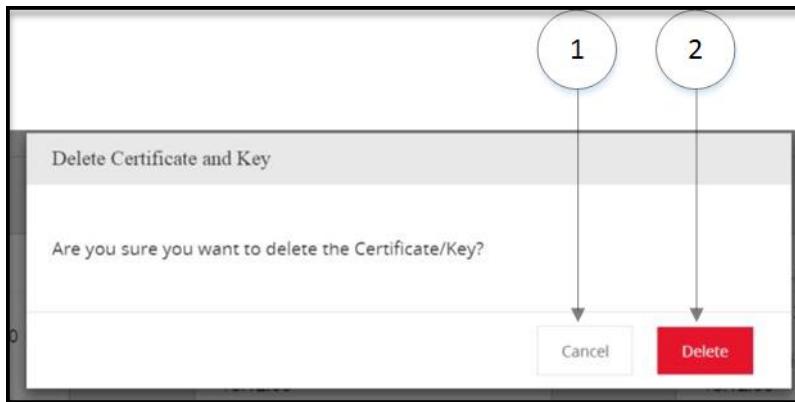


Figure 10-24: Delete Certificate

The following tables describes the available options:

Table 10-17: Delete Certificates screen UI options

Callout	Column/Textbox/Button	Description
1	Cancel	Cancel the process of deleting a certificate
2	Delete	Delete the certificate, key, or other files.

10.1.3.2 Tunnels tab

The Tunnels tab lets you define the DSG inbound communication channels. The changes made to Tunnels require cluster restart to take effect. You can either use the bundled default tunnels or create a tunnel based on your requirements.

The Tunnels tab is as seen in the following figure.

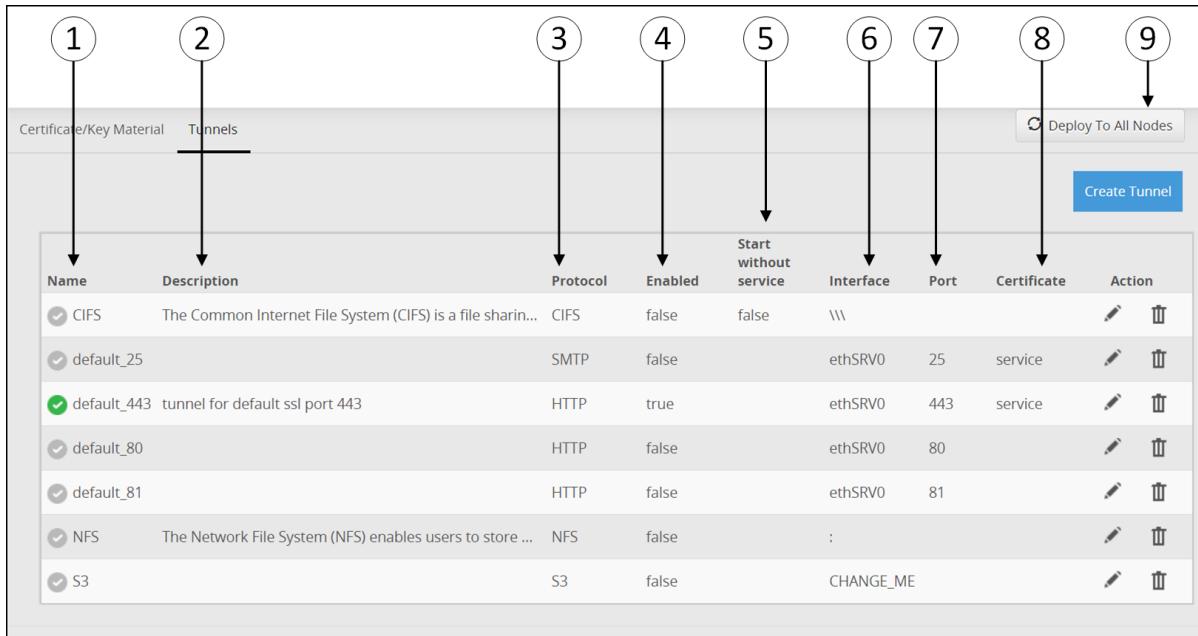


Figure 10-25: Tunnels tab of the ESA

The following table provides the description of the columns available on the Web UI.

Table 10-18: Tunnels Menu UI options

Callout	Column/Textbox	Description
1	Name	Unique tunnel name
2	Description	Unique description that describes port supported by the tunnel.
3	Protocol*	Protocol type that the tunnel supports.
4	Enabled	Status of the tunnel. Displays status as <i>true</i> , if the tunnel is enabled.
5	Start without service	Select to start the tunnel if no service is configured or if no services are enabled.
6	Interface**	IP address through which sensitive data enters the DSG.
7	Port	Port linked to the listening address.
8	Certificate	Certificate applicable to a tunnel.
9	Deploy to All Nodes	<p>Deploy the configurations to all the DSG nodes in the cluster.</p> <p>Note: Deploy can also be performed from the Cluster tab or <i>Ruleset</i> screen.</p> <p>Note: In a scenario where an ESA and two DSG nodes are in a cluster, by using the <i>Selective Tunnel Loading</i> functionality, you can load specific tunnel configurations on specific DSG nodes. Click Deploy to All Nodes to push specific tunnel configurations from an ESA to specific DSG nodes in a cluster.</p>

* -The available Type values are as follows:

- HTTP
- SMTP
- SFTP
- S3
- NFS
- CIFS

**-The available Listening Address options are as follows:

- ethMNG: The management interface on which the Web UI is accessible.
- ethSRV0: The service interface for communicating with an untrusted service.
- 127.0.0.1: The local loopback adapter.
- 0.0.0.0: The broadcast address for listening to all the available network interfaces over all IP addresses.
- Other: Manually add a listening address based on your requirements.

Note:

The service interface, *ethSRV0*, listens on port *443*. If you want to stop this interface from listening on this port, then edit the *default_443* tunnel and disable it.

For more information about editing the tunnel, refer to [Create or Edit Tunnel](#).

The following figure illustrates the actions for the *Tunnels* screen.

Name	Description	Protocol	Enabled	Start without service	Interface	Port	Certificate	Action
CIFS	The Common Internet File System (CIFS) is a file sharin...	CIFS	false	false	\\"			
default_25		SMTP	false		ethSRV0	25	service	
default_443	tunnel for default ssl port 443	HTTP	true		ethSRV0	443	service	
default_80		HTTP	false		ethSRV0	80		
default_81		HTTP	false		ethSRV0	81		
NFS	The Network File System (NFS) enables users to store ...	NFS	false	:				
S3		S3	false		CHANGE_ME			

Figure 10-26: Tunnels tab

The following table provides the available actions:

Table 10-19: Tunnels Menu Action items

Callout	Button	Description
1	Create Tunnel	Create a tunnel configuration as per your requirements.
2	Edit	Edit an existing tunnel configuration.
3	Delete	Delete an existing tunnel configuration.

The following sections describe the available options and the respective screen that appear in the Tunnels tab.

10.1.3.2.1 Create or Edit Tunnel

You can create tunnels for custom ports that are not predefined in the DSG using the Create Tunnel option in the Tunnels tab. The *Create Tunnel* screen is as seen in the following figure.

The screenshot shows the 'Create Tunnel' page under the 'Tunnels' tab. The 'Tunnel Details' section contains fields for Name (Sample DSG Tunnel), Description, Enabled (unchecked), Start without service (unchecked), and Protocol (HTTP). The 'Network Settings' section includes fields for Listening Interface (dropdown menu) and Port. The 'TLS/SSL Security Settings' section has a TLS Enabled checkbox. The 'Advanced Settings' section is a large empty text area. Callouts numbered 1 through 5 point to the Name, Description, Enabled, Start without service, and Protocol fields respectively.

Figure 10-27: Tunnels tab

The following table provides the description for each option available on the UI.

Table 10-20: Tunnels Menu UI options

Callout	Column/Textbox	Description
1	Name	Name of the tunnel.
2	Description	Unique description that describes port supported by the tunnel.
3	Enabled	Select to enable the tunnel. The check box is selected as a default. Uncheck the check box to disable the tunnel.
4	Start without service	Select to start the tunnel if no service is configured or if no services are enabled.
5	Protocol	Protocol type supported by the tunnel.

10.1.3.2.1.1 HTTP Inbound Settings

Based on the protocol selected, the dependent fields in the *Tunnel* screen vary. The following image illustrates the settings that are specific to the HTTP protocol.

The screenshot displays the 'HTTP Tunnel settings' configuration page. It includes sections for Network Settings and TLS/SSL Security Settings. Various fields are labeled with callout numbers 1 through 13, pointing to specific configuration options:

- Protocol:** HTTP (highlighted with a red box)
- Network Settings:**
 - Listening Interface*: Ex. eth0
 - Port*
- TLS/SSL Security Settings:**
 - TLS Enabled (checkbox) - Callout 3
 - Certificate - Callout 4
 - Cipher Suites: ECDH+AESGCM:DH+AESGCM:CDH+AES256:DH+AES256:ECDH+AES128 - Callout 5
 - TLS Mutual Authentication: CERT_NONE - Callout 6
 - CA Certificates: Ex:/opt/protegility/alliance/config/resources/ca.pem - Callout 7
 - DH Parameters: Ex:/opt/protegility/alliance/config/resources/dh_param.pem - Callout 8
 - ECDH Curve Name: Ex: prime256v - Callout 9
 - Certificate Revocation List: Ex:/opt/protegility/alliance/config/resources/(CRL_File).pem - Callout 10
 - Verify Flags: Ex: {OP_NO_COMPRESSION:true} - Callout 11
 - SSL Options: Ex: {OP_NO_COMPRESSION:true} - Callout 12
 - Advanced Settings - Callout 13

Figure 10-28: HTTP Tunnel settings

The options for the Inbound Transport Settings field in the *Tunnel Details* screen specific to the HTTP Protocol type are described in the following table.

Table 10-21: HTTP Inbound Settings

Callout	Column/Textbox/Button	Description
	Network Settings	
1	Listening Interface*	IP address through which sensitive data enters the DSG.
2	Port	Port linked to the listening address.
	TLS/SSL Security Settings	
3	TLS Enabled	Select to enable TLS features.
4	Certificate	Certificate applicable for a tunnel.
5	Cipher Suites	Colon separated list of Ciphers.
6	TLS Mutual Authentication**	<i>CERT_NONE</i> is selected as default. Use <i>CERT_OPTIONAL</i> to validate if a client

Callout	Column/Textbox/Button	Description
		<p>certificate is provided or <i>CERT_REQUIRED</i> to process a request only if a client certificate is provided.</p> <p>Note: If TLS mutual authentication is set to <i>CERT_OPTIONAL</i> or <i>CERT_REQUIRED</i>, then the CA certificate must be provided.</p>
7	CA Certificates**	<p>A CA certificate chain. This option is applicable only if the value client certificate is set to 1 (optional) or 2 (required).</p>
8	DH Parameters	<p>The <i>.pem</i> filename that includes the DH parameters. You can upload the <i>.pem</i> file from the <i>Certificate/Key Material</i> screen.</p> <p>Note: The <i>Diffie-Hellman (DH)</i> parameters define the way OpenSSL performs the DH Key exchange.</p>
9	ECDH Curve Name	<p>Supported curve names for the ECDH key exchange.</p> <p>The Elliptic curve Diffie–Hellman (ECDH) protocol allows key agreement and leverages elliptic-curve cryptography (ECC) properties for enhanced security.</p>
10	Certificate Revoke List***	<p>Path of the Certificate Revocation List (CRL) file.</p> <p>For more information about CRL error message that appears when a revoked certificate is sent, refer to the CRL error.</p>
11	Verify Flags***	<p>Set to one of the following operations to verify the CRL:</p> <ul style="list-style-type: none"> • VERIFY_DEFAULT • VERIFY_X509_TRUSTED_FIRST • VERIFY_CRL_CHECK_LEAF • VERIFY_CRL_CHECK_CHAIN <p>Note: The certificates are checked against the CRL file only for the inbound connections to the DSG node.</p>
12	SSL Options	<p>Set the required flags that reflect the TLS behavior at runtime. A single flag or multiple flags can be used.</p>

Callout	Column/Textbox/Button	Description
		<p>It is used to define the supported SSL options in the JSON format.</p> <p>For example, in the following JSON, TLSv1 and TLSv1.1 are disabled.</p> <pre>{ "options": ["OP_NO_SSLv2", "OP_NO_SSLv3", "OP_NO_TLSv1", "OP_NO_TLSv1_1"] }</pre> <p>The DSG supports TLS v1.2.</p>
13	Advanced Settings****	<p>Set additional advanced options for tunnel configuration, if required, in the form of JSON.</p> <p>Note: In a scenario where an ESA and two DSG nodes are in a cluster, by using the <i>Selective Tunnel Loading</i> functionality, you can load specific tunnel configurations on specific DSG nodes.</p>

* **The following Listening Interface options are available:**

- ethMNG: The management interface on which the DSG Web UI is accessible.
- ethSRV0: The service interface for communicating with an untrusted service.
- 127.0.0.1: The local loopback adapter.
- 0.0.0.0: The broadcast address for listening to all the available network interfaces.
- Other: Manually add a listening address based on your requirements.

** **The CA certificates are configured with settings available on the *Ruleset* screen.**

Client certificates can be requested at the tunnel and the RuleSet level for authentication. On the *Tunnels* screen, you can configure the *ca_reqs* parameter in the **Inbound Transport Settings** field to request the client certificate. Similarly, on the *Ruleset* screen, you can toggle the **Required Client Certificate** checkbox to enable or disable client certificates. Based on the combination of the options in the tunnel and the RuleSet, the server executes the transaction. If the certificate is incorrect or not provided, then server returns a *401 error* response.

The following table explains the combinations for the client certificate at the tunnel and the RuleSet level.

Table 10-22: TLS Mutual Authentication at Tunnel and RuleSet level

TLS Mutual Authentication (Tunnel Screen)	Required Client Certificate (Enable/Disabled) (Ruleset Screen)	Result
CERT_NONE	Disabled	The transaction is executed

TLS Mutual Authentication (Tunnel Screen)	Required Client Certificate (Enable/Disabled) (Ruleset Screen)	Result
	Enabled	The server returns a 401 error response.
CERT_OPTIONAL	Disabled	The transaction is executed
	Enabled	If the client certificate is provided, then transaction is executed. If the client certificate is not provided, then the server returns a 401 error response.
CERT_REQUIRED	Disabled	The transaction is executed
	Enabled	The transaction is executed

*** The Certificate Revocation List (CRL) and verifying information related to flags

The `ca.crl.pem` file includes a list of certificates that are revoked. Based on the flags that you provide in the `verify_flags` setting, SSL identifies certificate verification operations that need to be performed. The CRL verification operations can be `VERIFY_CRL_CHECK_LEAF` or `VERIFY_CRL_CHECK_CHAIN`.

When you try to access the DSG through HTTPS using such a revoked certificate, the DSG returns the following error message.

```
* Initializing NSS with certpath: sql:/etc/pki/nssdb
*   CAfile: ./ca.crt
*   CPath: none
* NSS: client certificate from file
*       subject: CN=dsg.client@protegility.com,O=Protegility,L=Stamford,ST=Connecticut,C=US
*       start date: Sep 01 12:42:04 2017 GMT
*       expire date: Sep 11 12:42:04 2018 GMT
*       common name: dsg.client@protegility.com
*       issuer: CN=protegility.com,O=Protegility,L=Stamford,ST=Connecticut,C=US
* NSS error -12270 (SSL_ERROR_REVOKED_CERT_ALERT)
* SSL peer rejected your certificate as revoked.
* Closing connection 0
curl: (58) SSL peer rejected your certificate as revoked.
```

Figure 10-29: Certificate Revoked error

****The advanced settings that can be configured for the HTTP Protocol.

Table 10-23: HTTP inbound settings

Options	Description	Default (if any)
idle_connection_timeout	Timeout set for an idle connection. The datatype for this option is seconds.	3600
max_buffer_size	Maximum value of incoming data to a buffer. The datatype for this option is bytes.	10240000
max_write_buffer_size	Maximum value of outgoing data to a buffer. The datatype for this option is bytes.	10240000
	<p>Note:</p> <p>This parameter is applicable only with REST streaming.</p>	
no_keep_alive	If set to <code>TRUE</code> , then the connection closes after one request.	false
decompress_request	Decompress the <code>gzip</code> request body	false

Options	Description	Default (if any)
chunk_size	Bytes to read at one time from the underlying transport. The datatype for this option is bytes.	16384
max_header_size	Maximum bytes for HTTP headers. The datatype for this option is bytes.	65536
body_timeout	Timeout set for wait time when reading request body. The datatype for this option is seconds.	
max_body_size	<p>Maximum bytes for the HTTP request body. The datatype for this option is bytes.</p> <p>Note: Though the DSG allows to configure the maximum body size, the response body size will differ and cannot be configured on the DSG.</p> <p>The response body size that the gateway will send to the HTTP client is dependent on multiple factors, such as, the complexity of the rule, transform rule configured in case you use regex replace, size of response received from destination, and so on.</p> <p>Note: If a request is sent to the client with the response body size greater than the value configured in the DSG, then the following response is returned and the DSG closes the connection: 400 Bad Request</p> <p>In earlier versions of the DSG, the DSG closed the connection and sent <i>200</i> as the response code.</p>	4194304
max_streaming_body_size	Maximum bytes for the HTTP request body when HTTP streaming with REST is enabled. The datatype for this option is bytes.	52428800
maximumBytes	Note: This field is not supported for the DSG 3.0.0.0 release and will be supported in a later DSG release.	
maximumRequests	Note: This field is not supported for the DSG 3.0.0.0 release and will be supported in a later DSG release.	

Options	Description	Default (if any)
thresholdDelta	<p>Note: This field is not supported for the DSG 3.0.0.0 release and will be supported in a later DSG release.</p>	
write_cache_memory_size	<p>For an HTTP blocking client sending a REST streaming request, the DSG processes the request and tries to send the response back. If the client type is blocking, then DSG will store the response to the memory till the <i>write_cache_memory_size</i> limit is reached. The DSG then starts writing to the disk.</p> <p>The file size is managed using the <i>write_cache_disk_size</i> parameter.</p> <p>The value for this setting is defined in bytes.</p>	<ul style="list-style-type: none"> • Min - 10485760 • Default - 52428800 • Max - 104857600
write_cache_disk_size	<p>Set the file size that holds the response after the <i>write_cache_memory_size</i> limit is reached while processing the REST streaming request sent by an HTTP blocking client.</p> <p>After the <i>write_cache_disk_size</i> limit is reached, the DSG starts writing to the disk.</p> <p>The data on the disk always exists in an encrypted format and the disk cache file is discarded after the response is sent.</p> <p>The value for this setting is defined in bytes.</p>	<ul style="list-style-type: none"> • Min - 52428800 • Default - 104857600 • Max - 314572800
additional_http_methods	Include additional HTTP methods, such as, PURGE, LINK, LINE, UNLINK, and so on.	
cookie_attributes	Add a new HTTP cookie to the list of cookies that the DSG accepts.	["expires", "path", "domain", "secure", "httponly", "max-age", "version", "comment", "priority", "samesite"]
compress_response	Compresses the response sent to the client if the client supports <i>gzip</i> encoding, i.e. sends <i>Accept-Encoding:gzip</i> .	false

10.1.3.2.1.1.1 Generating ECDSA certificate and key

The *dh_params* parameter points to a *.pem* file. The *.pem* file includes the DH parameters that are required to enable DH key exchange for improved protection without compromising computational resources required at each end. The value accepted by this field is the file name with the extension (*.pem*). The DSG supports both RSA certificates and Elliptic Curve Digital Signature Algorithm (ECDSA) certificates for the ECDHE protocol. The RSA certificates are available as default when the DSG is installed, while to use ECDSA certificates in the DSG, you must generate an ECDSA certificate and the related key. The following procedure explains how to generate the ECDSA certificate and key.

► To generate dhparam.pem file:

1. Set the SSL options in the Inbound Transport settings as given in the following example.
 - DH Parameters: */opt/protegility/alliance/config/dhparam/dhparam.pem*
 - ECDH Curve Name: *prime256v1*

- SSL Options: OP_NO_COMPRESSION
- From the ESA CLI Manager, navigate to **Administration > OS Console**.
 - Execute the following command to generate the *dhpam.pem* file.

```
openssl dhparam -out /opt/protegility/alliance/config/dhparam/dhparam.pem 2048
```

Note: Ensure that you create the dhparam directory in the given path. The path */opt/protegility/alliance/config/dhparam* is the location where you want to save the *.pem* file. The value 2048 is the key size.

- Execute the following command to generate the key.

Note: openssl genpkey -paramfile dhparam.pem -out dhkey.pem

The *ecdh_curve_name* parameter is the curve type that is required for the key exchange. The OpenSSL curves that are supported by DSG are listed in [Appendix E: Supported OpenSSL Curve Names and Options](#).

Note: You can configure additional inbound settings that apply to HTTP from the [Global Settings](#) page on the DSG Web UI.

10.1.3.2.1.2 SFTP Inbound Settings

Based on the protocol selected, the dependent fields in the *Tunnel* screen vary. The following image illustrates that settings specific to SFTP protocol.

The screenshot shows the 'Tunnel' configuration page for the SFTP protocol. The 'Protocol' dropdown is set to 'SFTP'. The 'Network Settings' section contains fields for 'Listening Interface*' (set to 'Ex. ethSRV0') and 'Port*' (set to '0'). The 'SSH Transport Security Options' section contains a field for 'Server Host Key Filename*' (set to 'Ex. service.key'). The 'Advanced Settings' section is a large text area for additional JSON options.

Figure 10-30: SFTP Tunnel Settings

The options specific to the SFTP Protocol type are described in the following table.

Table 10-24: SFTP Inbound Settings

Callout	Column/Textbox/Button	Description
	Network Settings	
1	Listening Interface*	IP address through which sensitive data enters the DSG.

Callout	Column/Textbox/Button	Description
2	Port	Port linked to the listening address.
	SSH Transport Security Options	SFTP specific security options that are mandatory. Select a paired server host key or provide the key path.
3	Server Host Key Filename	<p>Paired server host public key, uploaded through <i>Certificate/Key material</i> screen, that enables SFTP authentication.</p> <p>If the key includes an extension, such as *.key, enter the key name with the extension.</p> <p>For Files that are not uploaded to the resources directory, you must provide the absolute path along with the key name.</p> <p>Note: The DSG only accepts private keys that are not passphrase encrypted.</p>
4	Advanced Settings*	<p>Set additional advanced options for tunnel configuration, if required, in the form of JSON.</p> <p>Note: In a scenario where an ESA and two DSG nodes are in a cluster, by using the <i>Selective Tunnel Loading</i> functionality, you can load specific tunnel configurations on specific DSG nodes.</p>

* The available Listening Interface options are as follows:

- ethMNG: The management interface on which the Web UI is accessible.
- ethSRV0: The service interface for communicating with an untrusted service.
- 127.0.0.1: The local loopback adapter.
- 0.0.0.0: The broadcast address for listening to all the available network interfaces overall IP addresses.
- Other: Manually add a listening address based on your requirement.

**-The advanced settings that can be configured for SFTP Protocol.

Table 10-25: SFTP Inbound Settings

Options	Description	Default (if any)
idle_connection_timeout	Timeout set for an idle connection. The datatype for this option is seconds.	30
default_window_size	SSH Transport window size	2097152
default_max_packet_size	Maximum packet transmission in the network. The datatype for this option is bytes.	32768
use_compression	Toggle SSH Compression	True

Options	Description	Default (if any)
ciphers	List of supported ciphers	'aes128-ctr', 'aes256-ctr'
kex	Key exchange algorithms	'ecdh-sha2-nistp256', 'ecdh-sha2-nistp384', 'ecdh-sha2-nistp521', 'diffie-hellman-group16-sha512', 'diffie-hellman-group-exchange-sha256', 'diffie-hellman-group14-sha256', 'diffie-hellman-group14-sha1' 'curve25519-sha256@libssh.org', diffie-hellman-group-exchange-sha1, diffie-hellman-group1-sha1
digests	List of supported hash algorithms used in authentication.	'hmac-sha1'

The following snippet describes the example format for the SFTP Advanced settings:

```
{
  "idle_connection_timeout": 30,
  "default_window_size": 2097152,
  "default_max_packet_size": 32768,
  "use_compression": true,
  "ciphers": [
    "aes128-ctr",
    "aes256-ctr"
  ],
  "kex": [
    "diffie-hellman-group16-sha512"
  ],
  "digests": [
    "hmac-sha1"
  ]
}
```

Note:

In the next major release, the support for *diffie-hellman-group-exchange-sha1* and *diffie-hellman-group1-sha1* key exchange algorithms will be removed.

10.1.3.2.1.3 SMTP Tunnel

The DSG can perform data security operations on the sensitive data sent by an Simple Mail Transfer Protocol (SMTP) client before the data reaches the destination SMTP server.

Over the internet, SMTP is an Internet standard for sending emails. When an email is sent to anyone, the email is sent using an SMTP client to the SMTP server. For example, if an email is sent from *john.doe@xyz.com* to *jane.smith@abc.com*, the email first reaches the *xyz*'s SMTP server, then reaches *abc*'s SMTP server, before it finally reaches the recipient, *jane.smith@abc.com*.

The DSG intercepts the communication between the SMTP client and server and performs data security operations on sensitive data. The sensitive data residing in the email elements, such as subject of an email, body of an email, attachments, filename, and so on, are supported for the SMTP protocol:

When the DSG is used as an SMTP gateway, the Rulesets must use the *SMTP* service and the first child *Extract* rule must be *SMTP Message*.

The following image illustrates how the SMTP protocol is handled in the DSG. Consider an example where, *john.doe@xyz.com* is sending an email to *jane.smith@xyz.com*. The *xyz* SMTP server is the same for the sender and the recipient.

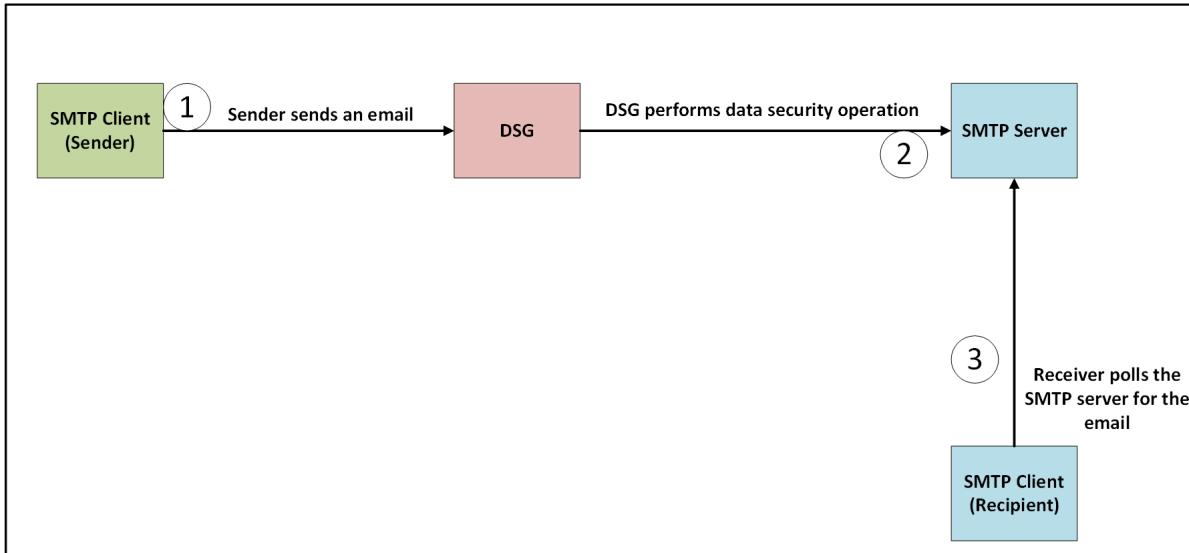


Figure 10-31: SMTP Protocol

1. The sender, *john.doe@xyz.com*, sends an email to the recipient, *jane.smith@xyz.com*. The *Subject* of the email contains sensitive data that must be protected before it reaches the recipient.
2. The DSG is configured with an SMTP tunnel such that it listens for incoming requests on the listening ports. The DSG is also configured with Rulesets such that an Extract rule extracts the *Subject* from the request. The Extract rule also defines a regex that extracts the sensitive data and passes it to the Transform rule. The Transform rule performs data security operations on the sensitive data.

The DSG forwards the email with the protected data in the *Subject* to the SMTP server.

3. The recipient SMTP client polls the SMTP server for any emails. The email is received and the sensitive data in the *Subject* appears protected.

The following image illustrates the settings specific to the SMTP protocol.

Setting	Value	Description
Protocol	SMTP	Configure the type of tunnel based on the desired application/transport protocol.
Listening Interface	ethSRV0	Listener binding IP address, hostname or NIC name; use 0.0.0.0 to listen on all NICs.
Port	25	Listening port number for the NIC.
Certificate		Server-side PKI certificate to enable TLS/SSL security.
Cipher Suites	ECDH+AESGCM:DH+AESGCM:ECDH+AES256:DH+AES256:ECDH+AES128	String in the textual OpenSSL cipher list format Read More
Advanced Settings		Set additional advanced options for tunnel configuration, if required, in the form of JSON in below textbox.

Figure 10-32: SMTP Tunnel Settings

The options specific to the SMTP Tunnel are described in the following table.

Table 10-26: SMTP Inbound Settings

Callout	Column/Textbox/Button	Description
	Network Settings	
1	Listening Interface*	Enter the service IP of the DSG, where the DSG listens for the incoming SMTP requests.
2	Port	Port linked to the listening address.
	Security Settings for SMTP	
3	Certificate	Server-side Public Key Infrastructure (PKI) certificate to enable TLS/SSL security.
4	Cipher Suites	Semi-colon separated list of Ciphers. For more information about Ciphers, refer to https://www.openssl.org/docs/man1.0.2/apps/ciphers.html .
5	Advanced Settings	Set additional advanced options for tunnel configuration, if required, in the form of JSON. Note: In a scenario where an ESA and two DSG nodes are in a cluster, by using the <i>Selective Tunnel Loading</i> functionality, you can load specific tunnel configurations on specific DSG nodes.

The *ssl_options* supported for the SMTP Tunnel are described in the following table.

Table 10-27: Inbound Transport Settings

Options	Description	Default
cert_reqs	Specifies whether a certificate is required for validating the SSL connection between the SMTP client and the DSG. The following values can be configured: <ul style="list-style-type: none"> • CERT_NONE: If the parameter is set to <i>CERT_NONE</i>, then the SMTP client certificate is not required for validating the SSL connection between the SMTP client and the DSG. • CERT_OPTIONAL: If the parameter is set to <i>CERT_OPTIONAL</i>, then the SMTP client certificate is not required for validating the SSL connection between the SMTP client and the DSG. The SMTP client certificate is validated only if it is provided. • CERT_REQUIRED: If the parameter is set to <i>CERT_REQUIRED</i>, then the SMTP client certificate is required for validating the SSL connection between the SMTP client and the DSG. 	<i>CERT_NONE</i>
ssl_version	Specifies the SSL protocol version used for establishing the SSL connection between the SMTP client and the DSG.	<i>PROTOCOL_SSLv23</i>



Options	Description	Default
ca_certs	Path where the CA certificates (in PEM format only) are stored.	n/a

* The following Listening Interface options are available:

- *ethMNG*: The management interface on which the DSG Web UI is accessible.
- *ethSRV0*: The service interface where the DSG listens for the incoming SMTP requests.
- *127.0.0.1*: The local loopback adapter.
- *0.0.0.0*: The broadcast address for listening to all the available network interfaces.
- *Other*: Manually add a listening address based on your requirements.

**-The advanced settings that can be configured for SMTP Protocol.

Table 10-28: SMTP Inbound Settings

Options	Description	Default (if any)
idle_connection_timeout	Timeout set for an idle connection. The datatype for this option is seconds.	30
default_window_size	SSH Transport window size	2097152
default_max_packet_size	Maximum packet transmission in the network. The datatype for this option is bytes.	32768

10.1.3.2.1.4 Amazon S3

Amazon Simple Storage Service (S3) is an online file storage web service. It lets you manage files through browser-based access as well as web services APIs. In DSG, the S3 tunnel is used to communicate with Amazon S3 cloud storage over the Amazon S3 REST API. The higher-layer S3 Service object, which sits above the tunnel object, configured at the RuleSet level is used to process file contents retrieved from S3.

A sample S3 tunnel configuration is shown in the following figure.

Tunnel Details

Name* Name for the tunnel, ex: DSG_HTTP.

Description Describe the purpose of the tunnel.

Enabled Select to enable the tunnel.

Protocol Configure the type of tunnel based on the desired application/transport protocol.

Bucket List Settings

Source Bucket Name* Bucket name as defined in AWS with specific folder where the files that need to be processed are available.
Ex. john.doe/incoming

Source File Name Pattern Filenames that match regex pattern provided here are processed.

Test Regex

Rename Processed Files

Match Pattern Regex logic with the source file to identify groups that can be used when defining replace value.
Ex. incoming/(.*)

Replace Value Value to append or name that will be used to rename the original source file based on the pattern provided and grouping defined in regex logic.
Ex. original/h1.done

Test Regex

Overwrite Target Object Select to overwrite a file in the bucket with the newly processed file with the same name.

Lock Files Bucket Name Name of the lock files folder, if you want the lock files to be stored in a separate bucket. If not defined, lock files are placed in the source bucket.
Ex. lockbucket

Interval Time in secs when the DSG node will poll AWS for pulling files. You can also provide a cron job, ex. * * * * *.

AWS Settings

AWS Access Key Id Access key id used to make secure protocol request to an AWS service API.

AWS Secret Access Key Secret access key related to the access key id.[Read More](#)

AWS Endpoint URL The complete URL that DSG as a client will connect to. Botocore library will by default connect to Amazon S3 bucket. This parameter should be configured only if you are using DSG to connect to other endpoint than S3 i.e. on premise S3.[Read Less](#)

Path to CA Bundle This specified path to CA certificate if endpoint is other than Amazon S3 bucket. Botocore library by default used SSL for connecting to S3 bucket. If you have installed S3 on premise using self signed certificate you can provide path to CA bundle.[Read Less](#)

Advanced Settings

Set additional advanced options for tunnel configuration, if required, in the form of JSON in below textbox.

Figure 10-33: Amazon S3 tunnel screen

Amazon S3 uses buckets to store data and data is classified as objects. Each object is identified with a unique key ID. Consider an example that *john.doe* is the bucket and *incoming* is a folder under *john.doe* bucket. Assuming the requirement is that files landing in the *incoming* folder should be picked up and processed by DSG nodes. The data pulled from the AWS online storage is available in the *incoming* folder under the source bucket. The Amazon S3 Service is used to perform data security operation on this data in the source bucket.

Note: The DSG supports four levels of nested folders in an Amazon S3 bucket.

After the rules are executed, the processed data may be stored in a separate bucket (e.g. the folder named *outgoing* under the same *john.doe* bucket), which is the target bucket. When the DSG nodes poll AWS for a file uploaded, whichever node accesses the file first places a lock on the file. You can specify if the lock files must be stored in a separate bucket or under the source bucket. If the file is locked, the other DSG nodes will stop trying to access the file.

If the data operation on a locked file fails, the lock file can be viewed for detailed log and error information. The lock files are automatically deleted if the processing completes successfully. The lock file in case of multiple folders can be seen as follows in the *lock* folder.

Consider the scenario where an incoming bucket contains two directories, *Folder1* and *Folder2*.

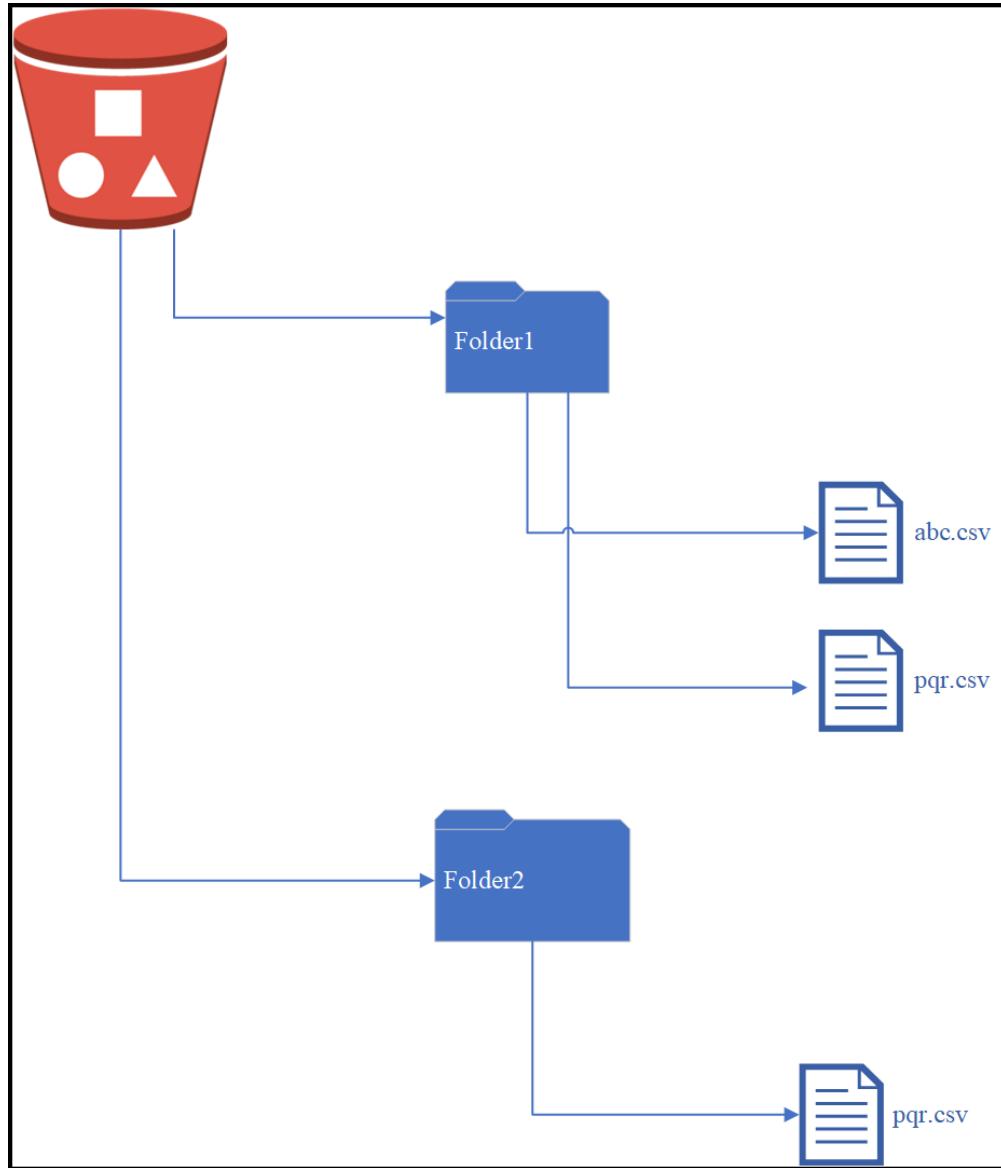


Figure 10-34: Scenario 1: S3 bucket with multiple files

The *Folder1* contains files *abc.csv* and *pqr.csv* while *Folder2* contains the file *pqr.csv*. When a DSG process picks a file for processing, it creates a lock file. In this case, a lock file *Folder1.hostname.pid.lock* is created. At the same time, when another DSG process attempts to pick the file *pqr.csv* for processing, it notices that a lock file for that particular folder is already created. This process has to wait for the *abc.csv* file to be processed before it proceeds ahead with the *pqr.csv* file. Thus, two files in a same folder cannot be simultaneously processed. However, if any DSG process attempts to process the file *pqr.csv* in *Folder2*, it is able to do so as the lock for *Folder2* does not exist.

Consider the scenario where files are nested in directories.

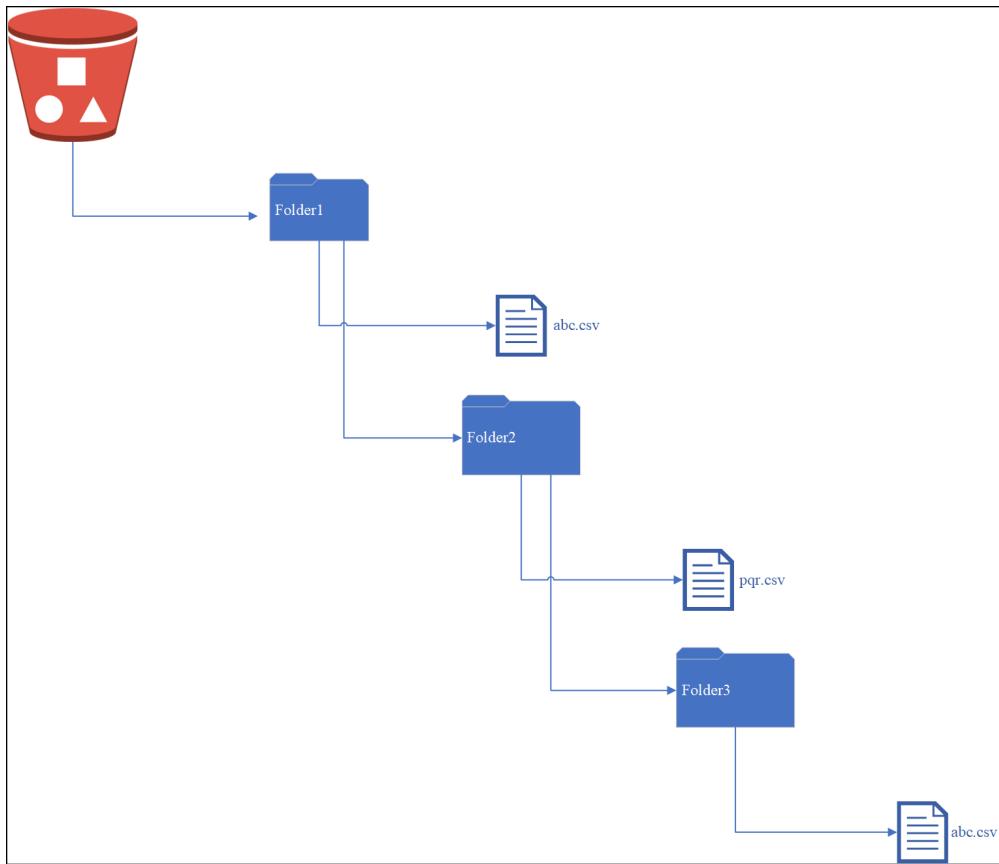


Figure 10-35: Scenario 2: S3 bucket with nested directories

When the file *pqr.csv* in *Folder2* is processed, a lock file is created as *Folder1.hostname.pid*. Similar to Scenario1, when another DSG process attempts to process the *abc.csv* file in *Folder1* it is not able to do so as a lock file already exists.

This issue can be mitigated by enabling the *enhanced-lock-filename* flag from the *features.json* file available in the **System > Files** on the DSG Web UI. You must edit this file and set the parameter as shown in the following snippet.

```
{
  features {
    "enhanced-lock-filename"
  }
}
```

When this flag is enabled, the lock files are created based on the directory and filename. For example, in scenario 1 the lock file are created as follows:

- If the *abc.csv* file of *Folder1* is processed, the lock file is created as *Folder1.abc.csv.<hostname>.pid.lock*.
- If the *pqr.csv* file of *Folder2* is processed, the lock file is created as *Folder1.Folder2.pqr.csv.<hostname>.pid.lock*.

In scenario 2, the lock files are created as follows:

- If the *abc.csv* file of *Folder1* is processed, the lock file is created as *Folder1.abc.csv.<hostname>.pid.lock*.
- If the *pqr.csv* file of *Folder2* is processed, the lock file is created as *Folder1.Folder2.pqr.csv.<hostname>.pid.lock*.
- If the *abc.csv* file of *Folder3* is processed, the lock file is created as *Folder1.Folder2.Folder3.abc.csv.<hostname>.pid.lock*.

This ensures that multiple processes can pick up files simultaneously without waiting for a file to be processed completely by a single process.

The following image illustrates the options available for an S3 tunnel.

Protocol: S3 (Configure the type of tunnel based on the desired application/transport protocol.)

Bucket List Settings

- Source Bucket Name***: Ex. john.doe/incoming (Bucket name as defined in AWS with specific folder where the files that need to be processed are available.)
- Source File Name Pattern**: (Test Regex) (Filenames that match regex pattern provided here are processed.)

Rename Processed Files

- Match Pattern**: (Test Regex) (Regex logic with the source file to identify groups that can be used when defining replace value.) Ex. incoming(“.”)
- Replace Value**: (Test Regex) (Value to append or name that will be used to rename the original source file based on the pattern provided and grouping defined in regex logic.) Ex. original\1.done

Overwrite Target Object: (Select to overwrite a file in the bucket with the newly processed file with the same name.)

Lock Files Bucket Name: Ex. lockbucket (Name of the lock files folder, if you want the lock files to be stored in a separate bucket. If not defined, lock files are placed in the source bucket.)

Interval: (Time in secs when the DSG node will poll AWS for pulling files. You can also provide a cron job, ex. “* * * * *”.)

AWS Settings

- AWS Access Key Id**: (Access key id used to make secure protocol request to an AWS service API.)
- AWS Secret Access Key**: (Secret access key related to the access key id. [Read More](#))
- AWS Endpoint URL**: (The complete URL that DSG as a client will connect to. BotoCore library will by default connect to Amazon S3 bucket. This parameter should be configured only if you are using DSG to connect to other endpoint than S3 i.e. on-premise S3. [Read Less](#))
- Path to CA Bundle**: (This specified path to CA certificate if endpoint is other than Amazon S3 bucket. BotoCore library by default uses SSL for connecting to S3 bucket. If you have installed S3 on-premise using self-signed certificate, you can provide the path to CA bundle. [Read Less](#))

Advanced Settings: (Set additional advanced options for tunnel configuration, if required, in the form of JSON in below textbox.)

Figure 10-36: S3 Tunnel Settings

The options specific to the S3 Protocol type are described in the following table.

Table 10-29: S3 Inbound Settings

Callout	Column/Textbox/Button	Description	Default(if any)
	Bucket List Settings		
1	Source Bucket Name	Bucket name as defined in AWS where the files that need to be processed are available. Note: The DSG supports four levels of nested folders in an Amazon S3 bucket.	
2	Source File Name Pattern	Filenames that match regex pattern provided here are processed.	
	Rename Processed Files	Regex logic for renaming processed file.	
3		Match Pattern	Regex logic with the source file to identify groups that can be used when defining replace value.

Callout	Column/Textbox/Button			Description	Default(if any)
4			Replace Value	Value to append or name that will be used to rename the original source file based on the pattern provided and grouping defined in regex logic.	
5		Overwrite Target Object		Set as true to overwrite a file in the bucket with the newly processed file with the same name. Note: For more information about how to define the buckets and objects that store the plaintext data and protected data, refer Amazon S3 Object .	
6		Lock Files Bucket		Name of the lock files folder, if you want the lock files to be stored in a separate bucket.	If not defined, lock files are placed in the source bucket.
7		Interval		Time in secs when the DSG node will poll AWS for pulling files. You can also specify a cron job expression. For more information about how to schedule cron jobs, refer to Cron documentation. Note: Cron job format is also supported to schedule jobs.	5 Note: If you use the cron job expression " * * * * * ", DSG will poll AWS at the minimum interval of one minute.
	AWS Settings				
8		AWS Access Key Id		Access key id used to make secure protocol request to an AWS service API. For more information about how to find the access key id and secret access key, refer to Amazon Web Service documentation.	
9		AWS Secret Access Key		Secret access key related to the access key id. The access key id and secret access key work together to sign into AWS and provide access to resources. For more information about how to find the access key id and secret access key, refer to Amazon Web Service documentation.	
10		AWS Endpoint URL		Specify the endpoint URL if it is other than the amazon S3 bucket. This parameter should only be configured if the user is using DSG to connect to other endpoint than amazon S3 bucket i.e. On-Premise S3, Google Cloud Bucket, and so on.	If not defined, the DSG will connect to Amazon S3 bucket.
11		Path to CA Bundle		Specify the path to CA bundle if the endpoint is other than Amazon S3 bucket. If the user have installed the S3 on-premise using the self signed certificate, then specify that path to CA bundle in this parameter.	If the endpoint URL is Amazon S3 bucket, then by default it uses SSL certificate to connect to S3 bucket.

Callout	Column/Textbox/Button	Description	Default(if any)
12	Advanced Settings*	<p>Set additional advanced options for tunnel configuration, if required, in the form of JSON in the following textbox.</p> <p>Note: In a scenario where an ESA and two DSG nodes are in a cluster, by using the Selective Tunnel Loading functionality, you can load specific tunnel configurations on specific DSG nodes.</p>	

* The advanced settings that can be configured for SFTP Protocol.

Table 10-30: S3 Inbound Settings

Options	Description	Default (if any)
SSECustomerAlgorithm	If server-side encryption with a customer-provided encryption key was requested, the response will include this header confirming the encryption algorithm used.	
SSECustomerKey	Constructs a new customer provided server-side encryption key.	
SSECustomerKeyMD5	If server-side encryption with a customer-provided encryption key was requested, the response will include this header to provide round-trip message integrity verification of the customer-provided encryption key.	
ACL	<p>Allows controlling the ownership of uploaded objects in an S3 bucket.</p> <p>For example, if ACL or Access Control List is set to "bucket-owner-full-control", new objects uploaded by other AWS accounts are owned by the bucket owner. By default, the objects uploaded by other AWS accounts are owned by them.</p>	

10.1.3.2.1.5 NFS/CIFS

The Network File System (NFS) enables users to store and access data from storage points such as disks and directories over a shared network. Though the files are accessed remotely, the behavior is same as when files are accessed locally. The NFS file system follows a client/server model, where the server is responsible for authentication and permissions, while the client accesses data through the local disk systems.

The Common Internet File System (CIFS) is a file sharing protocol for Windows OS-based systems.

A sample NFS/CIFS tunnel configuration is shown in the following figure.



Mount Network Settings

Protocol: NFS Configure the type of tunnel based on the desired application/transport protocol.

Mount Point*: 10.10.3.6:/home/NFSshare1 Define the path of the NFS mount point as seen in the NFS share. Ex. <IP Address or Hostname>:Mount_Path

Input Directory*: input Directory where the mount tunnel will search for new files. Ex. Input

Source File Name Pattern: Test Regex Filenames that match regex pattern provided here are processed.

Overwrite Target File: Set as true to overwrite a file in the mount location with the newly processed file with the same name.

Rename Processed Files

Match Pattern: (*.*)input/(.*.csv) Test Regex Regex logic with the source file to identify groups that can be used when defining replace value. [Read More](#)

Replace Value: \1output\2.done Test Regex Value to append or name that will be used to rename the original source file based on the pattern provided and grouping defined in regex logic. [Read More](#)

Trigger File

Trigger File Name Pattern*: %ctl Trigger file with this pattern, when placed inside input folder, will trigger processing of file name matching this pattern. '%ctl' here represents the input file which will be placed in the pattern to get trigger file name. [Read More](#)

Delete Trigger File: Select to delete trigger file , after the respective file gets processed.

Lock Files Directory: lock Directory where the lock files will be stored. If lock directory is not defined, the lock files will be stored in the same location as of the input file. Ex. Lock

Error Files Directory: error In case of an error during processing of a file, the file and its corresponding trigger file are moved to this directory. If this is not specified, the error files will remain in same location as of the input file. Ex. error

Error Files Extension: err Extension name to identify error file and its corresponding control file. In case it is not specified .err extension will be used. Ex. err

Mount Options

Mount Type: Soft If the remote server is unavailable, specifying Soft reports an error after server is unavailable beyond mount timeout and specifying Hard does not report an error until connection is interrupted.

Mount Timeout*: 10 Number in seconds after which an error is reported.

Options: Set other mount options for NFS/CIFS ,if required ,in key=value[,key=value] format

Ex. noLOCK, port=2049

Advanced Settings

: {"fileChunkSize": 4096} Set additional advanced options for tunnel configuration, if required, in the form of JSON in below textbox.

Figure 10-37: NFS/CIFS

Note: The Address format for an NFS tunnel is `<ip address/hostname>:<mount_path>` and for a CIFS tunnel is `\<ip address or hostname>\<share_path>`.

Consider an example NFS/CIFS server with folder structure that includes folders namely, `/input`, `/output`, and `/lock`. When a client accesses the NFS/CIFS server, the files are stored in the `/input` folder. The Mounted File System Out-of-Band Service is used to perform data security operation on the files in the `/input` folder. A source file is processed only when a corresponding trigger file is created and found in the `/input` folder.

Note: Ensure that the trigger file time stamp is greater than or equal to the source file time stamp.

After the rules are executed, the processed files can be stored in a separate folder, such as in this example, `/output`. When DSG nodes poll NFS/CIFS server for a file uploaded, whichever node accesses the file first places a lock on the file. You can specify if

the lock files must be stored in a separate bucket, such as `/lock` or under the source folder. If the file is locked, the other DSG nodes will stop trying to access the file.

If the data operation on a locked file fails, the lock file can be viewed for detailed log and error information. The lock files are automatically deleted if the processing completes successfully.

For NFS/CIFS, to allow multiprocessing, enable the `enhanced-lock-filename` flag from the `features.json` file available in the **System > Files** on the DSG Web UI. You must edit this file and set the parameter as shown in the following snippet.

```
{  
  features {  
    "enhanced-lock-filename"  
  }  
}
```

The lock file is created as follows:

```
<directory hierarchy>.<filename>.<hostname>.<process ID>.lock
```

For example, a file `test.csv` in the `Sample` directory that is placed in the `/incoming` folder generates a lock file as follows.

```
incoming.Sample.test.csv.<hostname>.<Process ID>.lock
```

10.1.3.2.1.5.1 NFS Tunnel

This section describes the options available for the NFS tunnel.

Caution: Ensure that the NFS share options are configured in the `exports` configuration file for each mount that the DSG will access. The `all_squash` option must be set to specify the `anonuid` and `anongid` with the user ID and group ID of the non-root user respectively.

This prevents the DSG from changing user and group permissions of the mount directories on the NFS server.

The following image illustrates the options available for an NFS tunnel.

Protocol NFS Configure the type of tunnel based on the desired application/transport protocol.

Mount Network Settings

- 1 Mount Point* Define the path of the NFS mount point as seen in the NFS share.
Ex. <IP Address or Hostname>:Mount_Path
- 2 Input Directory* Directory where the mount tunnel will search for new files.
- 3 Source File Name Pattern Test Regex Filenames that match regex pattern provided here are processed.
- 4 Overwrite Target File Set as true to overwrite a file in the mount location with the newly processed file with the same name.

Rename Processed Files

- 5 Match Pattern Test Regex Regex logic with the source file to identify groups that can be used when defining replace value.
Ex. (*.*)input/(.*.csv)
- 6 Replace Value Test Regex Value to append or name that will be used to rename the original source file based on the pattern provided and grouping defined in regex logic.
Ex. \1output\2.done

Trigger File

- 7 Trigger File Name Pattern* %ctl Control file with this pattern, when placed inside input folder, will trigger processing of file name matching this pattern. It will also be used as lock file , which will contain logs showing processing of that particular file.
Ex. %.ctl
- 8 Delete Trigger File Select to delete control file , after the individual file gets processed.
- 9 Lock Files Directory Directory where the lock files will be stored. If lock directory is not defined, the lock files will be stored in the same location as of the input file.
- 10 Error Files Directory Ex. error_dir In case of an error during processing of a file, the file and its corresponding trigger file are moved to this directory.If this is not specified,the error files will remain in same location as of the input file.
- 11 Error Files Extension err Extension name to identify error file and its corresponding control file.In case it is not specified .err extension will be used.
Ex. err

Mount Options

- 12 Mount Type Soft If the remote server is unavailable, specifying Soft reports an error after server is unavailable beyond mount timeout and specifying Hard does not report an error until connection is interrupted.
- 13 Mount Timeout* 60 Number in seconds after which an error is reported.
- 14 Options Ex. nolock,port=2049 Set other mount options for NFS/CIFS ,if required ,in key=value[,key=value] format

Advanced Settings Set additional advanced options for tunnel configuration,if required, in the form of JSON in below textbox.

Figure 10-38: NFS/CIFS Tunnel

The available fields for NFS tunnel are listed in following table.

Table 10-31: NFS Inbound Settings

Call out	Column/Textbox/Button	Description	Default (if any)
	Mount Network Settings		
1	Mount Point	The Address format for an NFS tunnel is <ip address/hostname>:<mount_path>.	

Call out	Column/Textbox/Button		Description	Default (if any)
2	Input Directory		The mount tunnel forwards the files present in this directory for further processing. This directory structure will be defined in the NFS/CIFS share.	
3	Source File Name Pattern		Regex logic for identifying the source files that must be processed.	
4	Overwrite Target File		Select to overwrite a file in the bucket with the newly processed file with the same name.	
	Rename Processed Files		Regex logic for renaming original source files after processed files are generated.	
5	Match Pattern		Exact pattern to match and filter the file.	
6	Replace Value		Value to append or name that will be used to rename the original source file based on the pattern provided and grouping defined in regex logic.	
	Trigger File		<p>File that triggers the rule. The rule will be triggered, only if this file is found to exist in the <i>input</i> directory.</p> <p>Files in the NFS/CIFS Share directory are not processed until the trigger criteria is met.</p> <p>Ensure that the trigger file is sent only after the files that need to be processed are placed in the source directory. After the trigger file is placed, you must touch the trigger file.</p>	
7	Trigger File Name Pattern		<p>Identifier that will be appended to each source file to create a trigger control file.</p> <p>For example, for a source file <i>abc.csv</i>, if you define the identifier as <i>%.ctl</i>, you must create a trigger file <i>abc.csv.ctl</i> to ensure that the source file is processed.</p> <p>Note: It is mandatory to provide a trigger file for each source file to ensure that it is processed. Files without a corresponding trigger file will not be processed.</p> <p>Note: The <i>*</i>, <i>/</i>, and <i>/</i> characters are not accepted as part of the trigger file pattern.</p>	
8	Delete Trigger File		Enable to delete the trigger file after the source file is processed.	

Call out	Column/Textbox/Button	Description	Default (if any)
9	Lock Files Directory	<p>Directory where the lock files will be stored. If this value is not provided as per the directory structure in the NFS/CIFS share, then the lock files will be stored in the mount point.</p> <p>If the lock directory is not defined, then the lock files are automatically placed in the <code>/input</code> directory.</p> <p>Note: Ensure that the lock directory name does not include spaces. The DSG will not process files under the lock directory that includes spaces.</p>	
10	Error Files directory	<p>Files that fail to process are moved to this directory. The lock files generated for such files are also moved to this directory.</p> <p>For example, the file is moved from the <code>/input</code> directory to the <code>/error</code> directory.</p>	
11	Error Files Extension	Extension that will be appended to each error file. If you do not specify an extension, then the <code>.err</code> extension will be used.	
	Mount Options	Parameters that will be used to mount the share.	
12		<p>Specify <i>Soft</i> if you want the mount point to report an error, if the server is unreachable after wait time crosses the <i>Mount Timeout</i> value.</p> <p>If you select this value as <i>Hard</i>, ensure that the <i>Interrupt Timeout</i> check box remains selected.</p>	
13		Mount Timeout	Number in seconds after which an error is reported. 60
14	Options	<p>Additional NFS options that can be provided as inbound settings.</p> <p>For example, <code>{"port":1234, "nolock", "nfsv3": 3}</code></p> <p>Caution: To enable enhanced security for the mounted share, it is recommended that the following options are set: <code>noexec, nosuid, nodev</code> where: • noexec: Disallow execution of executable binaries on the mounted file system.</p>	

Call out	Column/Textbox/Button	Description	Default (if any)
		<ul style="list-style-type: none"> • nosuid: Disallow creation of set user id files on the file system. • nodev: Disallow mounting of special devices, such as, USB, printers, etc. 	
15	Advanced Settings	<p>Set additional advanced options for tunnel configuration, if required, in the form of JSON in the <i>Advanced Settings</i> textbox.</p> <p>For example, <code>{"interval":5, "fileChunkSize": 4096}</code></p> <p>Note: In a scenario where an ESA and two DSG nodes are in a cluster, by using the Selective Tunnel Loading functionality, you can load specific tunnel configurations on specific DSG nodes.</p>	

10.1.3.2.1.5.2 CIFS Tunnel

This section describes the options available for the CIFS tunnel.

The following image illustrates the options available for a CIFS tunnel.

Note:

The **Start without service** field is not applicable for the CIFS tunnel settings.

Protocol CIFS CIFS Configure the type of tunnel based on the desired application/transport protocol.

Mount Network Settings

- 1 Mount Point* Define the path of the CIFS mount point as seen in the CIFS share.
Ex. \\<IP Address or Hostname>\Share_Path
- 2 Input Directory* Directory where the mount tunnel will search for new files.
Ex. input
- 3 Source File Name Pattern Filenames that match regex pattern provided here are processed.
- 4 Overwrite Target File Set as true to overwrite a file in the mount location with the newly processed file with the same name.

Rename Processed Files

- 5 Match Pattern Test Regex Regex logic with the source file to identify groups that can be used when defining replace value.[Read More](#)
Ex. (.*)input/(.*.csv)
- 6 Replace Value Test Regex Value to append or name that will be used to rename the original source file based on the pattern provided and grouping defined in regex logic.[Read More](#)
Ex. \\1archive\\2.done

Trigger File

- 7 Trigger File Name Pattern* Test Regex Trigger file with this pattern, when placed inside input folder, will trigger processing of file name matching this pattern.%' here represents the input file which will be placed in the pattern to get trigger file name.[Read More](#)
Ex. %.ctl
- 8 Delete Trigger File Select to delete trigger file , after the respective file gets processed.
- 9 Lock Files Directory Directory where the lock files will be stored. If lock directory is not defined, the lock files will be stored in the same location as of the input file.
Ex. Lock
- 10 Error Files Directory In case of an error during processing of a file, the file and its corresponding trigger file are moved to this directory.If this is not specified,the error files will remain in same location as of the input file.
Ex. error
- 11 Error Files Extension Extension name to identify error file and its corresponding control file.In case it is not specified .err extension will be used.
Ex. err

Mount Options

- 12 User Name User name that will be used to authenticate with the server.
- 13 Password Password for the CIFS share.
- 14 Confirm Password Retype Password.
- File Mode Permissions to override default file mode, if server does not support CIFS Unix extensions.
- Directory Mode Permissions to override default directory mode, if server does not support CIFS Unix extensions.
- Read Write Permissions Enable read-write permission for the mount point.
- No Permission Enable to disallow client from performing permission checks. Remote permissions and UIDs will still be visible, but they will not be enforced locally.
- Options Set other mount options for NFS/CIFS ,if required ,in key=value[key=value] format
Ex. noserverino

Advanced Settings Set additional advanced options for tunnel configuration,if required, in the form of JSON in below textbox.

Figure 10-39: CIFS Tunnel

The available fields for the CIFS tunnel are listed in following table.

Caution:

The following mount options are deprecated for CIFS:

- File Mode
- Directory Mode
- Read Write Permissions
- No Permission
- Options

Table 10-32: CIFS Tunnel

Call out	Column/Textbox/Button			Description	Default(if any)
	Mount Network Settings				
1		Mount Point		The Address format for a CIFS tunnel is <ip address or hostname> <share_path>.	
2		Input Directory		The mount tunnel forwards the files present in this directory for further processing. This directory structure will be defined in the CIFS share.	
3		Source File Name Pattern		Filenames that match the regex pattern provided here are processed.	
4		Overwrite Target Object		Select to overwrite a file in the bucket with the newly processed file with the same name.	
		Rename Processed Files		Regex logic for renaming original source files after processed files are generated.	
5			Match Pattern	Exact pattern to match and filter the file.	
6			Replace Value	Value to append or name that will be used to rename the original source file, based on the pattern provided and grouping defined in regex logic.	
		Trigger File Name		<p>File name that triggers the rule. The rule will be triggered, only if this file is found in the <i>trigger</i> directory.</p> <p>Files in the CIFS Share directory are not processed until the trigger criteria is met.</p>	
7			Trigger File Name Pattern	<p>Identifier that will be appended to each source file to create a trigger control file.</p> <p>For example, for a source file <i>abc.csv</i>, if you define the identifier as <i>%.ctl</i>, you must create a trigger file <i>abc.csv.ctl</i> to ensure that the source file is processed.</p> <p>Note: It is mandatory to provide a trigger file for each source file to ensure that it is processed.</p>	

Call out	Column/Textbox/Button			Description	Default(if any)
				<p>Files without corresponding trigger file will not be processed.</p> <p>Note: The *, /, and / characters are not accepted as part of the trigger file pattern.</p>	
8		Delete Trigger File		Enable to delete the trigger file after the source file is processed.	
9		Lock Files Directory		<p>Directory where the lock files will be stored. If this value is not provided as per directory structure in the CIFS share, then the lock files will be stored in the mount point.</p> <p>If the lock directory is not defined, then the lock files are automatically placed in the <code>/input</code> directory.</p>	
10		Error Files directory		<p>Files that fail to process are moved to this directory. The lock files generated for such files are also moved to this directory.</p> <p>For example, the file is moved from the <code>/input</code> directory to the <code>/error</code> directory.</p>	
11		Error Files Extension		Extension that will be appended to each error file. If you do not specify an extension, then the <code>.err</code> extension will be used.	
		Mount Options		Parameters that will be used to mount the share.	
12			User Name	User name that will be used to authenticate with the server.	
13			Password	<p>Password for the CIFS username.</p> <p>Note: After the tunnel is saved, the password cannot be read back from the UI. The UI shows an encrypted password. The password maybe be reset, if required.</p>	
14			Confirm Password	Re-enter the CIFS password.	
15		Advanced Settings		Set additional advanced options for tunnel configuration, if required, in the form of JSON in the <i>Advanced Settings</i> textbox.	

Call out	Column/Textbox/Button	Description	Default(if any)
		<p>For example, <code>{"interval":5, "fileChunkSize": 4096}</code></p> <p>Note: In a scenario where an ESA and two DSG nodes are in a cluster, by using the Selective Tunnel Loading functionality, you can load specific tunnel configurations on specific DSG nodes.</p>	

Important:

The CIFS tunnel does not support kerberos authentication.

10.1.3.2.2 Delete Tunnel

You can delete existing tunnels using the Delete option in the Tunnels tab. The *Delete Tunnel* screen is shown in the following figure.



Figure 10-40: Delete Tunnel screen

The following table provides the description for each option available on the UI.

Table 10-33: Tunnels Menu UI options

Callout	Column/Textbox/Button	Description
1	Cancel	Cancel the process of deleting a tunnel.
2	Delete	Delete the existing tunnel from the Tunnels tab.

10.1.3.2.3 Edit Tunnel

You can edit any existing tunnels configuration using the Edit option in the Tunnels tab. The *Edit Tunnel* screen is as seen in the following figure.

Edit Tunnel Tunnels are instances of server-side transport/application protocol stacks in DSG.

Tunnel Details

- Name***: default_443 (Name for the tunnel, ex: DSG_HTTP.)
- Description**: tunnel for default ssl port 443 (Describe the purpose of the tunnel.)
- Enabled** Select to enable the tunnel.
- Start without service** Start the tunnel if no service is configured or if no services are enabled
- Protocol**: HTTP (Configure the type of tunnel based on the desired application/transport protocol.)

Network Settings

- Listening Interface***: ethSRV0 (Listener binding IP address, hostname or NIC name; use 0.0.0.0 to listen on all NICs. [Read More](#))
- Port***: 443 (Listening port number for the NIC.)
- Firewall rules may require adjustment to allow incoming connections from this particular port.

TLS/SSL Security Settings

- TLS Enabled** Select to enable TLS for secure Tunnel communication.

Advanced Settings

(Set additional advanced options for tunnel configuration, if required, in the form of JSON in below textbox.)

Buttons: Cancel, Update

Figure 10-41: Update Tunnel screen

This screen is like the *Create Tunnel* screen described above

After you edit the required field, you can click Update to save your changes.

10.1.4 Global Settings

The *Global Settings* screen lets you configure settings that affect a DSG node globally. You can configure debug options, global protocol settings, and Web UI settings that impact the DSG.

The following image illustrates the UI options on the Global Settings tab.

Debug Global Protocol Stack Web UI

Log Settings (System wide Debug configuration for DSG.)

Stats Log Settings (Enable stats logging to get information about the connections established and closed for any service on all or individual DSG nodes in a cluster.)

Admin Interface (Set the configurations to enable communication across DSG nodes for administrative queries.)

Global Learn Mode settings (Configuration settings for Learn Mode, a feature used for inspecting application protocol messages and their step by step processing in PCG. Settings specified here maybe overridden by Service/RuleSet specific configuration.)

Long Running Routines Tracing (Settings related to diagnostic tracing of long-running routines.)

Buttons: Deploy To All Nodes (1), (2), (3)

Figure 10-42: Global Settings UI

The following table provides the description for each of the available RuleSet options:

Table 10-34: Global Settings UI options

Callout	Icon	Column/Textbox/Button	Description
1		Deploy to All Nodes	<p>Deploy the configurations to all the DSG nodes in the cluster.</p> <p>Note: Deploy can also be performed from the Cluster tab.</p>
			<p>Note: In a scenario where an ESA and two DSG nodes are in a cluster, by using the <i>Selective Tunnel Loading</i> functionality, you can load specific tunnel configurations on specific DSG nodes. Click Deploy to All Nodes to push specific tunnel configurations from an ESA to specific DSG nodes in a cluster.</p>
2		Expand	Expand the subtab and view available options.
		Collapse	Collapse the subtab to hide the available options.
3		Edit	Edit the available options in the subtab.

10.1.4.1 Debug tab

The Debug tab lets you configure log settings, Learn mode settings, and set configurations that enable administrative queries.

The screenshot shows the 'Debug' tab in the Protegility Data Security Gateway Web UI. The interface is organized into several sections:

- Log Settings:** Set system wide log settings for all or individual DSG nodes in a cluster. Includes a dropdown for 'Log Level' set to 'Warning'.
- Stats Log Settings:** Enable stats logging to get information about the connections established and closed for any service on all or individual DSG nodes in a cluster. Shows 'Logging' status as 'Enabled'.
- Admin Interface:** Set the configurations to enable communication across DSG nodes for administrative queries. Includes fields for 'Listening Address*', 'Listening Port*', 'SSL Certificate', 'SSL Certificate Key', 'Client CA Certificate', 'Client Certificate', 'Client Certificate Key File', 'Common Name', 'OpenSSL Cipher Lists', and 'SSL Options'.
- Global Learn Mode settings:** Configuration settings for Learn Mode, a feature used for inspecting application protocol messages and their step by step processing in PCG. Settings specified here maybe overridden by Service/RuleSet specific configuration. Includes fields for 'Enabled', 'Exclude Payload Types', 'Exclude Content-Type', 'Include Resource', 'Include Content-Type', and 'Free Disk Space Threshold'.
- Long Running Routines Tracing:** Settings related to diagnostic tracing of long-running routines. Includes fields for 'Enabled' and 'Timeout'.

Figure 10-43: Debug tab settings

The following table provides information about fields in the Debug tab.

Table 10-35: Debug tab

Sub tab	Fields	Description
Log Settings	Log Level	<p>Set the logging level at the node level.</p> <p>Note:</p> <p>If the log level is updated from the Global Settings screen, then the log facility-related settings are removed from the <code>gateway.json</code> file. The user must manually add the log facility-related settings to the <code>gateway.json</code> file.</p>

Sub tab	Fields	Description
		For more information about the log facility-related settings, refer to the section Additional Configurations using the gateway.json file .
Admin Interface	Listening Address	<p>Listening address for the admin tunnel. The DSG listens for requests such as learn mode settings that are sent through the admin tunnel.</p> <p>Admin tunnel is a system tunnel that lets you send administrative requests to individual DSG nodes.</p>
	Listening Port	Listening port for the admin tunnel.
	SSL Certificate	The DSG admin certificate to authenticate inbound requests.
	SSL Certificate Key	Paired DSG admin key used with the admin certificate.
	Client CA Certificate	<p>Client CA certificate (<i>.pem</i>) file against which the client certificate will be validated.</p> <p>Note: The supported Client CA certificate file format is <i>.pem</i>.</p>
	Client Certificate	<p>Client certificate (<i>.pem</i>) file that is required for establishing communication between the ESA-DSG nodes and the DSG-DSG nodes.</p> <p>Note: The supported Client CA certificate file format is <i>.pem</i>.</p>
	Client Certificate Key File	Paired client certificate key.
	Common Name	<p>Common name defined in the client certificate.</p> <p>Note: Ensure that the Common Name (CN) defined in the client certificate matches the name defined in this field.</p>
	OpenSSL Cipher Lists	Semi-colon separated list of Ciphers.
	SSL Options	Options you must set for successful communication between the ESA-DSG nodes and the DSG-DSG nodes.
Stats Log Settings	Stats Logging Enabled	Enable stats logging to get information about the connections established and closed for any service on all or individual DSG nodes in a cluster.
Global Learn Mode Settings*	Enabled	Select to enable Learn Mode at node level.
	Exclude Payload Types	Resources matching this regex patterns are excluded from the Learn Mode logging.
	Exclude Content-Type	Protocol messages with Content-type headers are excluded from the Learn Mode logging.

Sub tab	Fields	Description
	Include Resource	Resources matching this regex pattern are included in the Learn Mode logging.
	Include Content-Type	Protocol messages with Content-type headers are included in the Learn Mode logging.
	Free Disk Space Threshold	Minimum free disk space required so that Learn Mode feature remains enabled. The feature is automatically disabled, if free disk space falls below this threshold. You must enable this feature manually, if it has been disabled.
Long Running Routines Tracing	Enabled	Enable stack trace for processes that exceed the defined timeout.
	Timeout	Define value in seconds to log a stack trace of processes that do not process smoothly in a given timeout. The default value is 20 seconds.

* - Settings provided in these fields can be overwritten by the settings provided at Service/Ruleset level.

10.1.4.2 Global Protocol Stack tab

Apart from the settings that you configure for each service type, some settings affect all services that relate to a protocol type.

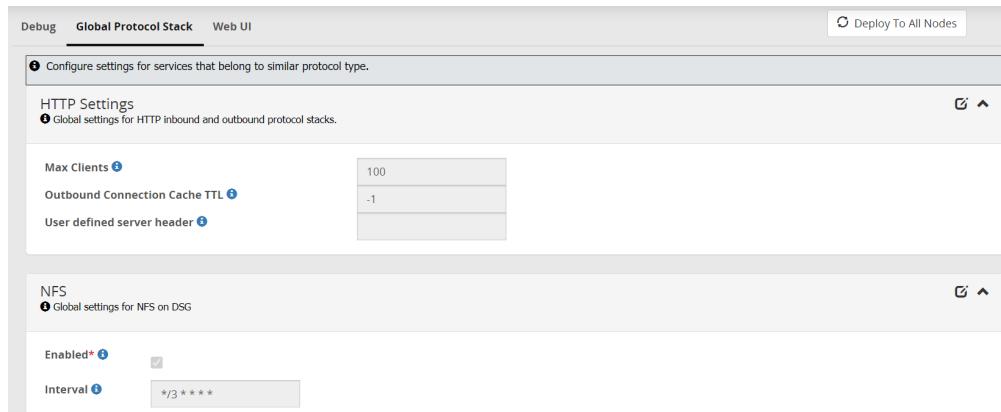


Figure 10-44: Global Protocol Stack

The following table provides information about the fields in the Global Protocol Stack tab.

Table 10-36: Global Protocol Stack options

Sub tab	Fields	Description	Default
HTTP	Max Clients	Set the maximum number of concurrent outbound connections every gateway process can establish with each host.	The default value for this setting is 100.
	User defined server header	If you want to change the value of the server header in an application response, then you can use this parameter.	

Sub tab	Fields	Description	Default
	Outbound Connection Cache TTL	In situations where you want to keep a TCP connection persistent beyond the default limit of inactivity that the firewall allows, you must configure this setting to a timeout value. The timeout value must be defined in seconds.	-1 This value indicates that the feature is disabled. The connection remains active and stored in cache until the DSG node is restarted.
NFS	Enabled	Set as true to enable the NFS tunnel and service.	
	Interval	Time in seconds when the DSG node will poll the NFS server for fetching files. You can also specify a cron job expression. For more information about how to schedule cron jobs, refer to the cron documentation. Note: The Cron job format is also supported to schedule jobs.	Note: If you use the cron job expression " * * * * * ", then the DSG will poll the NFS server at the minimum interval of one minute.

10.1.4.3 Web UI tab

The Web UI tab lets you configure additional settings that impact how the UI is displayed.

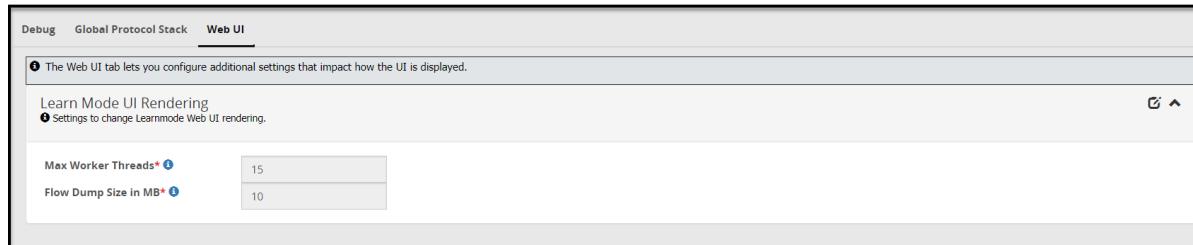


Figure 10-45: Web UI Tab

The following table provides information about fields in the Web UI tab.

Table 10-37: Web UI options

Sub tab	Fields	Description	Default
Learn Mode UI Performance Settings	Max Worker Threads	Maximum worker threads that would render learnmode flow dumps on screen.	15
	Flow Dump Filesize	The rules displayed in the <i>Learn mode</i> screen and the payload message difference are stored in a separate file in DSG. If the payloads and rules in your configuration are high in volume, you can configure this file size.	10 MB

10.1.5 Tokenization Portal

After you set up a cluster between the ESA and multiple DSG nodes, the policies are deployed on respective DSG nodes. Each policy has related data elements, data stores, and roles. You can use the Tokenization Portal menu to examine protection or unprotection of data when a protection data element is used.

The Tokenization Portal provides an interface where you can select the data security operation you want to perform, along with the DSG node, data elements available in the policies deployed at that node, and an external IV value. Every protection operation performed through the Tokenization Portal is recorded as an event in Forensics.

To access the test utilities, in the browser, enter https://<ESA_IP_Address>/TokenizationPortal.

Caution: Before you access the Tokenization Portal, ensure that the following pre-requisites are met:

- Ensure that any user who wants to access the test utilities must be granted the *Cloud Gateway User* and *Policy User* permissions.
- Ensure that the ESA where you are accessing Tokenization Portal is a part of the cluster.
- Ensure that the policy on the ESA is deployed to all the DSG nodes in the cluster.
- Ensure that the policy is *synchronized* across all the ESAs in the cluster.

The following image illustrates the UI options on the Tokenization Portal tab.

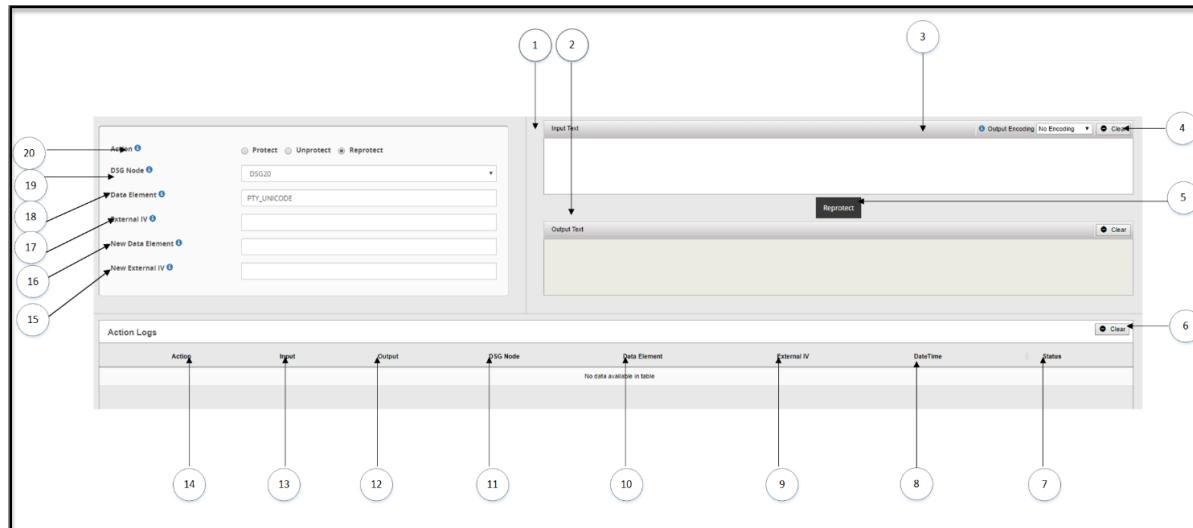


Figure 10-46: Tokenization Portal

The following table provides the description for each of the available Tokenization tab options:

Table 10-38: Tokenization Portal Menu UI Columns

Callout	Column/Textbox/Button	Sub-Columns	Description
1	Input Text		Enter the data you want to protect or unprotect.
2	Output Text		Transformed data, either protected or unprotected based on operation selected.
3	Output Encoding*		Select the type of encoding you want to use. Though the Output Encoding option is part of the Input Text area, remember that encoding is applied to Input data during

Callout	Column/Textbox/Button	Sub-Columns	Description
			protection and reprotection, and output data during unprotection.
4	Clear		Clear the text in the Input Text or Output Text box.
5	Unprotect/Protect/Reprotect		Click to perform security operation on the Input Text. You can either Protect, Unprotect, or Reprotect Data. Note: This option changes to Protect, Unprotect, or Reprotect based on the Action selected for data security operation.
6	Clear		Clear the text in the Input Text or Output Text box.
	Action Logs		Logs related to the data protection or unprotection are displayed under this area. These logs are cached in the browser session and are not persisted in the system.
7		Status	Displays  if data security operation was successful.
8		DateTime	Date and time when the data security operation was performed.
9		External IV	External IV value that was used for the data security operation.
10		Data Element	Data element used.
11		DSG Node	The DSG node where the data security operation was performed.
12		Output	Transformed data.
13		Input	Input data.
14		Action	Data security operation performed.
15	New External IV		New external IV value that will be used along with the protect or unprotect algorithm to create more secure encrypted data. Note: This field applies to the Reprotect option only.
16	New Data Element		New data element that will be used to perform data security operation. Note: This field applies to the eprotect option only.
17	External IV		External IV value that will be used along with the protect or unprotect algorithm to create more secure encrypted data.

Callout	Column/Textbox/Button	Sub-Columns	Description
18	Data Element		<p>Data element that will be used to perform data security operation.</p> <p>Note: If an encryption data element is used to protect sensitive data, then the output encoding type should be selected to protect the data successfully.</p>
19	DSG Node		The DSG node where the data security operation will be performed.
20	Action		Data security operation, Protect, Unprotect, or Reprotect that you want to perform.

* - The available encoding options are as follows:

- **Output Encoding:** The field appears when action selected is either Protect or Reprotect.
- **Input Encoding:** The field appears when action selected is Unprotect.

10.1.6 Additional Configurations using the `gateway.json` file

The **Global Settings** tab provides information about the different configurations, that when set, are enforced across all the DSG nodes. In addition to the setting that are part of the **Global Settings** tab, the `gateway.json` file also includes additional settings.

The `gateway.json` file includes configurations, such as, setting the log levels, enabling learn mode, and so on.

Caution:

It is recommended that any settings that must be changed, are edited on the ESA and then pushed to the DSG nodes in the cluster. To access the `gateway.json` file, on the ESA Web UI, navigate to **Settings > System**.

The configurations that can be controlled using the `gateway.json` file are displayed in the following code snippet.

```
{
  "admin": {
    "certificateFilename": "admin.pem",
    "certificateKeyFilename": "admin.key",
    "ciphers": "ECDH+AESGCM:DH+AESGCM:ECDH+AES256:DH+AES256:ECDH+AES128:DH+AES:RSA+AESGCM:RSA+AES:!aNULL:!MD5:!DSS:!SSLv2:!SSLv3!TLSv1!TLSv1.1",
    "clientCACertificateFileName": "ca.pem",
    "clientCertificateFileName": "admin_client.pem",
    "clientCertificateKeyFileName": "admin_client.key",
    "commonName": "protegilityClient",
    "listenAddress": "ethMNG",
    "listenPort": 8585,
    "ssl_options": "{\"cert_reqs\": \"CERT_REQUIRED\"}"
  },
  "globalProtocolStackSettings": {
    "http": {
      "connection_cache_ttl": -1,
      "include_hostname_in_header": true,
      "max_body_size": 4194304,
      "max_header_size": 8192
    }
  }
}
```



```

        "max_clients": 100,
        "max_streaming_body_size": 52428800
    }
},
"globalUDFSettings": {
    "blocked_methods": [ "eval", "exec", "dir", "__import__", "memoryview" ],
    "blocked_modules": [ "pip", "install", "commands", "subprocess", "popen2", "sys",
                         "os", "platform", "signal", "asyncio" ]
},
"learnModeDefault": {
    "enabled": false,
    "excludeContentType": "\\\bcss|image|video|svg\\\\b",
    "excludeResource": "\\\.(css|png|gif|jpg|ico|woff|ttf|svg|eot)(\\\\?|\\\\b)",
    "freeDiskSpaceThreshold": 1024000000
},
"log": {
    "logFacility": [
        {
            "enabled": true,
            "facilityName": "Tunnel",
            "logLevel": "Information"
        },
        {
            "enabled": false,
            "facilityName": "DiskBuffer",
            "logLevel": "Debug"
        }
    ],
    "logLevel": "Warning",
},
"longRunningRoutinesTracing": {
    "enabled": false,
    "timeout": 20
},
"mountManager": {
    "enabled": true,
    "interval": "*/3 * * * *"
},
"stats": {
    "enabled": false
}
}
}

```

The following table provides the description for each of the available configurations.

Note:

Some of the settings in this file are also available at the service level in the Rule definition.

Table 10-39: gateway.json configurations

Configuration	Sub-Configuration	Description	Default Value
log		Settings to control logging level	
	logLevel	<p>Set the logging level.</p> <p>The available logging levels, in increasing order of logged details displayed, are as follows:</p> <ul style="list-style-type: none"> • Warning • Info • Debug • Verbose 	Warning
	logFacility	Set the logging level for the following modules:	



Configuration	Sub-Configuration	Description	Default Value
		<ul style="list-style-type: none"> • RuleSet • Services • Tunnel • DiskBuffer • Admin 	
	checkErrorLogAfterCount	<p>This parameter decides the trimming factor that is a part of the error metrics. You can set this value in the range of -1 to 1000.</p> <p>If the value set is greater than -1 and the log size of the error metrics is greater than 4k, then it will trim the <i>error_metrics</i> in such a way that all the parameters will be displayed accurately and only the row number information will be trimmed.</p> <p>If the log size is not exceeding 4k, then the error metrics will be displayed as is.</p> <p>If the value is set to -1 and the log size of error metrics is greater than 4k, then all the characters after the 4k limit will be trimmed from the log file.</p> <p>Note: By default, this parameter is not included in the <i>gateway.json</i> file. To configure this parameter, user must manually add the <i>checkErrorLogAfterCount</i> in the <i>gateway.json</i> file.</p>	1000
mountManager		Settings related to CIFS/NFS mounts	
	enabled	Enable or disable mount management	true
	interval	<p>Time in seconds when the DSG node will poll the CIFS/NFS shares for pulling files. You can also specify a cron job expression.</p> <p>For more information about how to schedule cron jobs, refer to Cron documentation.</p> <p>Note: The cron job format is also supported to schedule jobs.</p> <p>Note:</p>	3

Configuration	Sub-Configuration	Description	Default Value
		If you use the cron job expression " <code>* * * * *</code> ", then the DSG will poll the CIFS/NFS shares at the minimum interval of one minute.	
admin		Settings related to the <i>admin</i> tunnel are listed here. The DSG uses the internal <i>admin</i> tunnel to communicate with the ESA and the other DSG nodes.	
	listenAddress	<p>Listening interface name, typically <i>ethMNG</i></p> <p>Caution: The default <i>listenAddress</i> must not be modified.</p>	<i>ethMNG</i>
	listenPort	Listening port for the listening interface	8585
	certificateFilename	<p>Admin tunnel certificate file name with the <i>.pem</i> extension</p> <p>Note: The default certificates and keys are set after the DSG is installed.</p> <p>For more information about handling default certificates and keys, refer to the section Default Certificates and Keys.</p>	<i>admin.pem</i>
	certificateKeyFilename	<p>Admin tunnel key file name with the <i>.key</i> extension</p> <p>Note: The default certificates and keys are set after the DSG is installed.</p> <p>For more information about handling default certificates and keys, refer to the section Default Certificates and Keys.</p>	<i>admin.key</i>
	ciphers	Colon separated list of Ciphers	ECDH+AESGCM:DH+AESGCM:EC DH+AES256:DH+AES256:ECDH+A ES128:DH+AES:RSA+AESGCM:RSA+AES:!aNULL:!MD5:!DSS
	clientCACertificateFilename	Admin tunnel CA certificate filename with the <i>.pem</i> extension	<i>ca.pem</i>

Configuration	Sub-Configuration	Description	Default Value
		<p>The default certificates and keys are set after the DSG is installed.</p> <p>For more information about handling default certificates and keys, refer to the section Default Certificates and Keys.</p>	
	clientCertificateFilename	<p>Admin tunnel Client certificate file name with the .pem extension</p> <p>Note: The default certificates and keys are set after the DSG is installed.</p> <p>For more information about handling default certificates and keys, refer to the section Default Certificates and Keys.</p>	<i>admin_client.pem</i>
	clientCertificateKeyfilename	<p>Admin tunnel Client key file name with the .key extension</p> <p>Note: The default certificates and keys are set after the DSG is installed.</p> <p>For more information about handling default certificates and keys, refer to the section Default Certificates and Keys.</p>	<i>admin_client.key</i>
	commonName	<p>Common name as defined while creating the admin tunnel client certificates</p> <p>Note: The default common name is set after the DSG is installed.</p> <p>For more information about common name, refer to the section Default Certificates and Keys.</p>	ProtegilityClient
	ssl_options	<p>Set the SSL options that you want to enforce</p> <p>Note: It is recommended that for secure communication between the ESA</p>	\"cert_reqs\":\"CERT_REQUIRED\"

Configuration	Sub-Configuration	Description	Default Value
		and the DSG, the <i>ssl_options</i> are not modified.	
learnModeDefault		Settings for the Learn Mode	
	enabled	Enable or disable Learn Mode on the DSG node	true
	excludeResources	Values in the field are excluded from the Learn Mode logging.	\.(css png gif jpg ico woff ttf svg eot)(\ \?\\b)
	excludedContentType	Content type specified in the field is excluded from the Learn Mode logging.	\bcss image video svg\b
	freeDiskSpaceThreshold	Minimum free disk space required so that the Learn Mode feature remains enabled. The feature is automatically disabled, if free disk space falls below this threshold. If the setting is disabled, then you must enable this feature manually. The datatype for this option is bytes.	1024000000
globalUDFSettings		Settings that apply to any rules defined with custom UDF implementation for a DSG node. For more information about the listed UDF settings, refer to the section User Defined Functions (UDF) .	
	blocked_modules	List of vulnerable modules that cannot be used in UDF definitions.	"pip", "install", "commands", "subprocess", "popen2", "sys", "os", "platform", "signal", "asyncio"
	blocked_methods	List of vulnerable methods that cannot be used in UDF definitions.	"eval", "exec", "dir", "__import__", "memoryview"
globalProtocolStackSettings (http)		Settings for incoming HTTP requests management.	
	max_clients	Sets the maximum number of concurrent outbound connection per gateway process for each host separately.	100
	include_hostname_in_header	By default, the hostname will be visible in response header. You can set the parameter to false, to remove the hostname from the response header.	true
	connection_cache_ttl	Timeout value that you must configure upto which an HTTP request connection persists. The following values can be set: <ul style="list-style-type: none">• <value> - Set the value in seconds• 0 - Set to disable caching• -1 - Set to enable caching	-1

Configuration	Sub-Configuration	Description	Default Value
	max_body_size	Maximum bytes for the HTTP request body. The datatype for this option is bytes.	4194304
	max_streaming_body_size	Maximum bytes for the HTTP request body when REST with streaming is enabled. The datatype for this option is bytes.	52428800
longRunningRoutinesTracing		Settings for stack trace of processes that exceed the defined timeout.	
	enabled	Enable or disable tracing	false
	timeout	Define the value in seconds to log a stack trace of processes that do not process easily in the given timeout interval. The datatype for this option is seconds.	20

Chapter 11

Overview of Sub-Clustering

11.1 Sub-Clustering FAQs

In a TAC where ESA and DSGs are setup, the configuration files are pushed from the ESA to the DSG nodes on the cluster. However, in versions prior to DSG 3.0.0.0, only a single copy of the gateway configurations could be pushed to the DSG nodes. The ability to push specific rulesets to specific nodes in the cluster was unavailable. From v3.0.0.0, this limitation has been eliminated by introducing the sub-clustering feature.

Sub-clustering allows the user to create separate clusters of the DSG nodes in a TAC. All the nodes in the sub-cluster contain the same rulesets. This enables the user to maintain different copies of rulesets for different sub-clusters. Sub-clusters can be used to define various Lines-of-Business (LOB) of the organization. The user can then create logical node groups to deploy the rulesets on different DSG nodes (LOBs) successfully.

For example, if an *XYZ* company is spread across the globe with multiple LOBs, then they can use sub-clustering feature to deploy the configurations to a particular node group (i.e. LOB1, LOB2, LOB3, and so on).

The following image illustrates how the sub-clustering feature is implemented for the DSG nodes.



Figure 11-1: Deployment for LOB1

The [figure](#) depicts three node groups LOB1, LOB2, LOB3. Consider LOB1, that caters to only *HTTP* and *S3* services. The *common* is the service that is shared among all the LOBs. This common service includes the protection profiles, unprotection profiles, and so on, that can be used by the user. Only the tunnels that are used for the enabled services will be loaded. The other tunnels will not be loaded. Thus, for LOB1, only the *HTTP* and *S3* tunnels will be loaded.

Perform the following steps to push the configurations to the LOB1 node group.

1. Add four DSG nodes from the cluster page and set the node group as *LOB1*.
For more information about adding a node and node group to the cluster, refer to the section [Adding a Node to the Cluster](#).
2. Enable the *Office 365* and *S3 service1* services. These services will be pushed to the LOB1 node group. Disable the *NFS service 1*, *SFTP service*, *restapi*, *Adlabs*, *salesforce*, and *S3 service 2* services
3. Select **LOB1** to push the rulesets to all the four DSG nodes in the LOB1 node group.

Note: For more information about deploying the configurations to node groups, refer to the section [Deploying the Configurations to Node Groups](#).

The following figure describes a sample use case for sub-clustering.

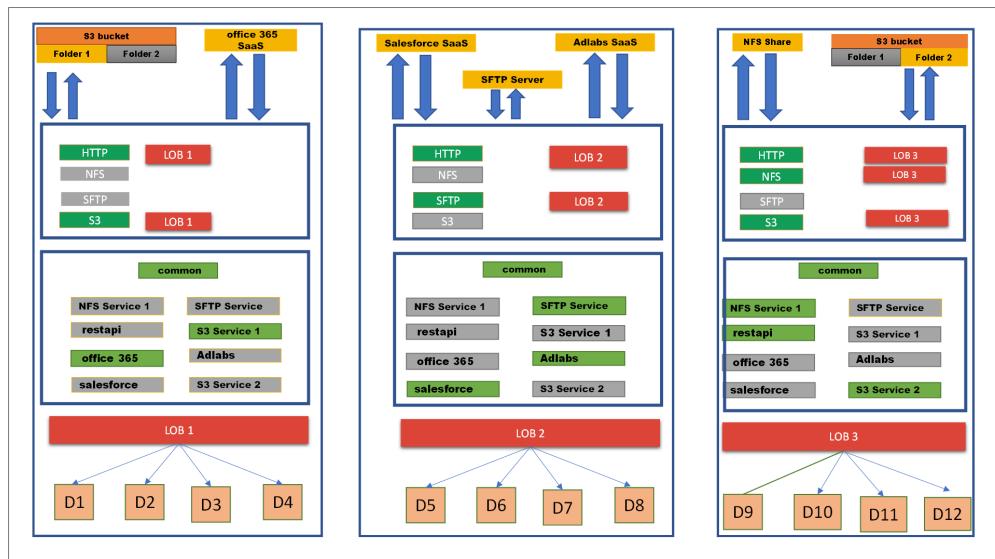


Figure 11-2: Use case for sub-clustering

As shown in [figure](#), consider LOB1, LOB2, and LOB3 are different lines of business that belong to an *XYZ* company. Each LOB are as follows:

- The LOB1 will use the *S3 bucket's folder 1* and *office 365 SaaS* services. This LOB is assigned to nodes D1, D2, D3, and D4.
- The LOB2 will use the *Salesforce SaaS*, *SFTP server*, and *Adlabs SaaS* services. This LOB is assigned to nodes D5, D6, D7, and D8.
- The LOB3 will use the *NFS share* and *S3 bucket's folder 2* services. This LOB is assigned to nodes D9, D10, D11, and D12.

All other services in the **RuleSet** page will be disabled to deploy the configurations to LOB1 node group.

Important Notes

- The sub-clustering feature can only be used when the DSG node is added from the **Cluster** screen of the ESA Web UI. It is recommended to add the node to the cluster only from this screen. While adding a node, a node group can be assigned to the DSG node. If a node group is not assigned, then a *default* node group is assigned to that DSG node.

For more information about adding the DSG node from cluster page, refer to the section [Adding a Node to the Cluster](#).

- The tunnels, certificates/keys, and gateway configuration files are common to all the DSGs in the cluster.
- If the user is using the Selective Tunnel Reloading feature with sub-clustering, then ensure that you prefix *dsg_* in the node group name while setting the tunnel configuration.
- For DSG Appliances, rulesets are deployed based on the node groups that are mapped to it.
- For DSG containers, the user can use CoP export API to export the configurations for a particular node group and then deploy it to the containers. This is achieved by passing the Node Group as a parameter to the export API.

For more information about CoP export API, refer to the section [Appendix H: CoP Export API for deploying the CoP\(Containers Only\)](#).

11.1 Sub-Clustering FAQs

Questions	Answers
What if I have to change the node group assigned to a DSG node?	If you have to change the node group of a node, then login to the ESA Web UI, navigate to Cloud Gateway > Cluster , then on the node click the Actions drop down list and select Change Groups option. Specify the required node group name and save it.
What if I have to change the node group on multiple DSG nodes at a time?	If you have to change the node group of multiple DSG nodes at a time, then login to the ESA Web UI, navigate to Cloud Gateway > Cluster , then select the check box given for a individual node on which the node group must be changed, click the Actions drop down list, and select Change Groups on Selected Nodes option. Specify the node group name and save the changes.
From where should be the DSG nodes added to the cluster?	The DSG node must be only added from the Cluster page. Login to ESA Web UI, navigate to Cloud Gateway > Cluster , click Actions drop down list and select Add Node option. For more information about adding a node to the cluster, refer to the section Add Node .
What if while adding a node to cluster, the deployment node group is not specified?	If the deployment node group is not specified, then by default it will get assigned to the default node group.
Can the DSG node be assigned to different node groups at a time?	No, the DSG node can be assigned to only one node group at a time. If required, then you can change the node group but at a time the node will be associated to one node group.
What would happen if you add the DSG node from the CLI or TAC?	It is not recommended to add the DSG node from the CLI or TAC. The sub-clustering feature would not work with all the functionalities.
Can we deploy the configurations to multiple node groups?	Yes, if you have different node groups and you click Deploy to Node Groups option on the Cluster tab or Ruleset screen then it will show all the node groups created. Select the check box of the node groups to which the configurations must be pushed.
How to configure the services in the Ruleset page, to push it to a particular node group?	Enable the required services and deploy it to the intended node group. <div style="background-color: #e0f2e0; padding: 5px;"> Note: Disable all the services that are not intended to be pushed on the node group. </div>
Can I have separate node groups as LOB1, lob1, or any combination of letters for this node group name?	All the combination of letters of the node group name is considered as same and it will be displayed in the lowercase.



Questions	Answers
Can we deploy the configuration to the node group without providing the tag name or description?	Yes, the tag name and description are not the mandatory fields. If you do not provide the tag name, then the configuration version will be untagged.
What would happen if the node group has a recently deployed configuration and you are assigning that node group to a DSG node?	In this scenario, the recently deployed configuration for that node group will be redeployed to the DSG node.

Chapter 12

Implementation

[12.1 Network Setup](#)

[12.2 Configuring Ruleset](#)

[12.3 Forwarding logs to SIEM systems](#)

[12.4 Rolling Out to Production](#)

This section describes the post configuration steps involved in implementing Data Security Gateway (DSG) including RuleSets Configuration, Migration, Testing and Production Rollout.

The configuration of a Ruleset would be subject to the data and how it is exchanged between the systems communicating through DSG. A common SaaS deployment example is followed and additional generic examples, where applicable, are provided.

The process involves the following steps:

1. Network Setup

- Domain Name System (DNS)
- Connectivity
- Upload or generate SSL Certificates
- Configure Tunnels

2. Configuring Ruleset

- Create a Service under Ruleset
- Use Learn Mode to find message carrying sensitive data
- Create a Profile under Service and define rules to intercept the message and transform it

3. Optional: Forwarding logs to external SIEM systems

4. Rolling Out to Production

The implementation section will continue to follow the prototypical organization, Biloxi Corp, example, an enterprise that chose to use a SaaS called [ffcrm.com](#). The [crm.example.com](#) story is followed to configure Ruleset profile.

12.1 Network Setup

As part of setting up the CRM, you must set up domains names and ssl certificates.

12.1.1 Domain Name System

The CRM SaaS selected by Biloxi Corp is accessible through the public domain name [www.ffcrm.com](#). To integrate the gateway in front of the CRM, the consumers need to be directed to the gateway's load balancer. A public domain like [www.ffcrm.com](#) is owned by a 3rd party otherwise the necessary changes could have been made so it gets resolved to address of the gateway's load balancer. Therefore, a domain name, [www.ffcrm.biloxi.com](#), is used to internally represent the external [www.ffcrm.com](#) one.

DNS would then be configured such that all sub domain names (i.e. **www.ffcrm.biloxi.com** or **anything.ffcrm.biloxi.com**) will always point to the same gateway's load balancer address.

The public domain name **www.ffcrm.com** will be accessible to Biloxi Corp users but that concern is addressed by preventing them from successfully authenticating with the service should they attempt to bypass the gateway, intentionally or by error.

In a lab or testing environment, one can alternatively modify their local system hosts file to achieve a similar goal. This approach only effects the local machine and does not support wild card configuration, hence each domain name must be specified explicitly.

For example: 2.10.1.53 ffcrm.biloxi.com www.ffcrm.biloxi.com

This part is completed when all SaaS domain names resolve to your gateway's load balancer IP Address.

12.1.2 Network Connectivity

Biloxi Corp is following network segregation best practices using network firewalls. Thus, communication between systems in different parts of the network is strictly controlled.

Depending on where the gateway is installed, some of the firewall's configuration need to be modified so that the gateway cluster is reachable and accessible to the consumers and that the public domain **www.ffcrm.com** is accessible to the gateway cluster.

Administrative web interface and ssh access to ESA and the gateway cluster may require adjusting network firewall settings.

Verify that the network is properly configured to allow users connectivity to the gateway cluster. This can be validated using tools like ping and telnet. In addition, you'll need to confirm the connectivity between the gateway nodes themselves and the public SaaS system.

12.1.3 SSL Certificates

The load balancer in front of the gateway cluster at Biloxi Corp is already equipped with the appropriate SSL Certificate however another certificate needs to be generated to secure the connectivity between the load balancer and the gateway cluster. You could generate the certificate elsewhere and add it to the list along with its key using the upload button.

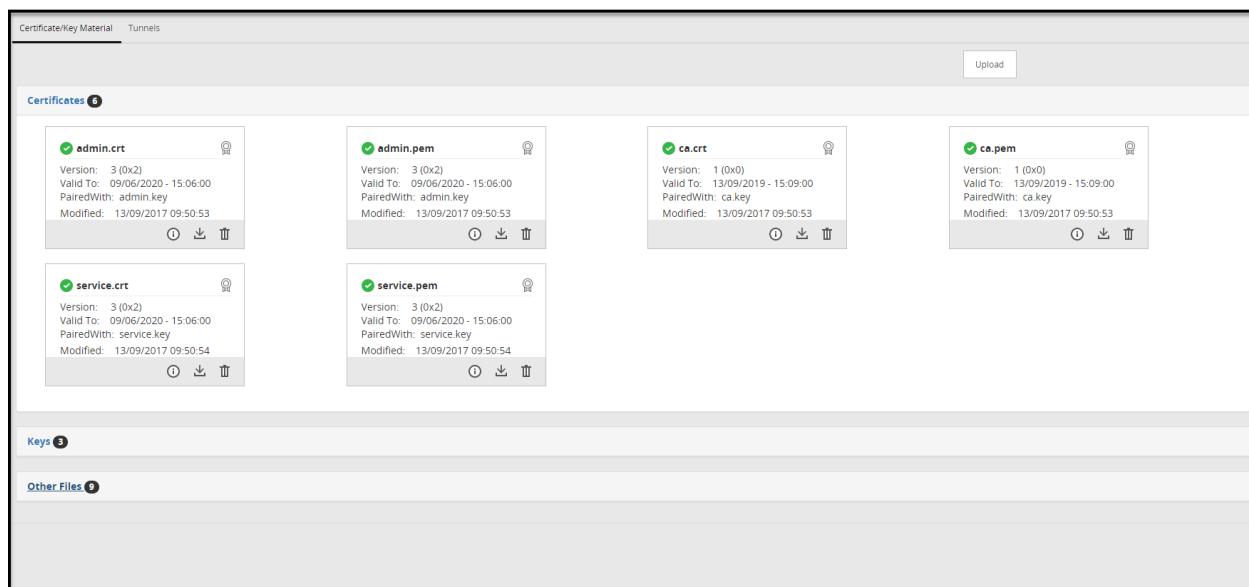


Figure 12-1: Generate Certificate screen

12.2 Configuring Ruleset

This section explains steps to create rules and how the rule trajectory can be tracked with Learn Mode.

12.2.1 Creating a Service under RuleSet

At this point it is not known how sensitive data is exchanged with the CRM application. To find out the details, the Learn mode functionality is used, which is controlled at the Service level.

Note: Before you begin creating services, ensure that you read the Best Practices for defining services.

Using the main menu navigate to **Cloud Gateway > RuleSet > RuleSet** and click the **Add New Service** link to create a new service. After giving it an appropriate name and description, the new HTTP Gateway Service with one or more tunnels needs to be associated, through which the **www.fferm.com** application network message exchange will be coming. Use the (+) icon next to the Tunnels and Hostnames fields to add and associate the service with the default_443 tunnel and the internal hostname to external forwarding address mapping.

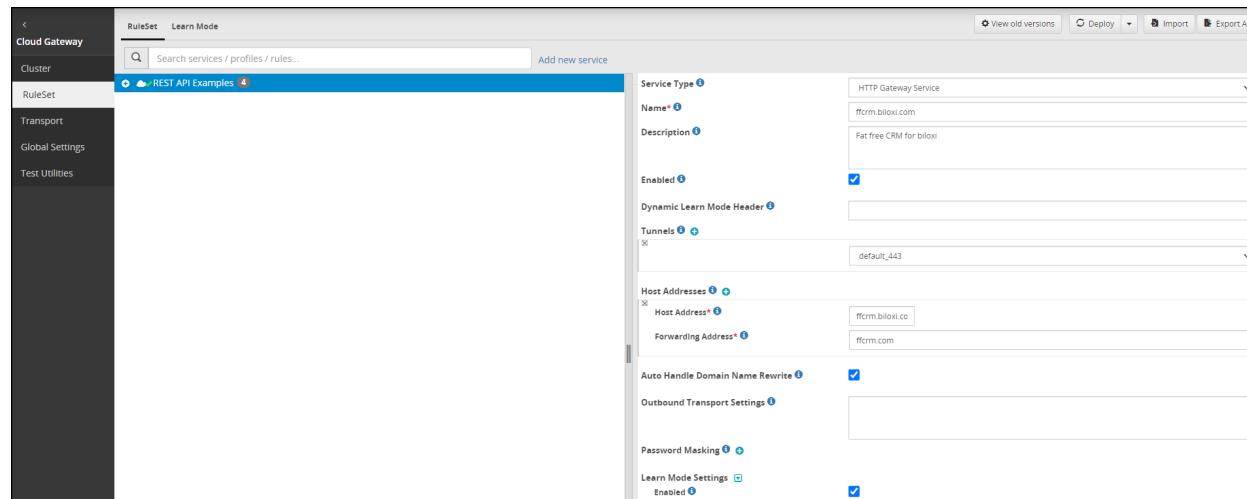


Figure 12-2: Create RuleSet screen

Note that the hostname is how DSG identifies which service applies to an incoming message.

The transaction metric logging feature available at the service level provides you the ability to log detailed metrics, such as, number of protect operations, length of protected data, service used, and so on. These details can be logged as part of the *gateway.log* whenever a protect operation HTTP request is made to the DSG.

For more information about transaction metrics options, refer to [Transaction Metrics logging](#).

The sample metrics as seen in the log file are as follows:

```
{
  "auth_cache_hit": 1,
  "auth_end_time": "2021-02-12 19:38:10.306866+00:00",
  "auth_start_time": "2021-02-12 19:38:10.306774+00:00",
  "auth_user_name": "test",
  "client_ip": "████████",
  "dsg_version": "1.0.0",
  "end_time": "2021-02-12 19:49:29.313049+00:00",
  "http_method": "POST",
  "http_reason_phrase": "OK",
  "http_status_code": 200,
  "learn_mode_enabled": false,
  "local_port": 45498,
  "node_hostname": "dsg241",
  "node_pid": 11144,
  "request_uri": "https://restapi:8099/dynamic/streaming",
  "service_name": "IRX REST Service",
  "start_time": "2021-02-12 19:38:10.306635+00:00",
  "total_time": "0:11:19.006414",
  "transformations": {
    "data_protection": {
      "data_elements": [
        {
          "data_element_name": "TE_A_N_S13_L0R0_Y",
          "len_protect": 219447,
          "num_protect": 24383
        },
        {
          "data_element_name": "TE_UA_N_S13_L0R0_Y",
          "len_protect": 1284680,
          "num_protect": 121915
        }
      ]
    }
  },
  "tunnel_name": "HTTP_8099",
  "user_name": "user1"
}
```

Figure 12-3: Transaction Metrics logged

Restart the gateway cluster for this change to take effect.

12.2.1.1 Trusting Self-Signed Certificates for an Outbound Communication

Next, Learn-mode is used to find the messages carrying sensitive data, examine their structure and design rules to protect it before it is forwarded on to www.ffcrm.com. For TLS-based outbound communications in DSG, it is expected that the server or SaaS uses a certificate that is signed by a trusted certification authority. In some cases, self-signed certificates are used for outbound communications. For such cases, the DSG is required to trust server's certificate to enable TLS-based outbound communications.

► Perform the following steps to trust self-signed certificates in the ESA node.

1. In the ESA Web UI, navigate to **Settings > System > File Upload**.
2. Click **Browse** to upload the self-signed certificates.

3. From the ESA CLI Manager, navigate to **Administration > OS Console**.

Note: Ensure that the certificate is in PEM (ASCII BASE64) format. This may be done by inspecting the certificate file using the cat command to verify whether the contents of the certificate are in readable (BASE64 encoded) or in a binary format.

4. If the contents of the certificate are in binary format (DER), execute the following command to convert the contents to PEM (BASE64 encoding), else proceed to [Step 7](#).

```
openssl x509 -inform der -in <certificate name>.cer -out <certificate name>.pem
```

5. Create a directory *opt/protegility/alliance/config/outbound_certs/* if it does not exist.
6. Create a file **trusted.cer** under */opt/protegility/alliance/config/outbound_certs* if it does not exist.
7. Copy the contents of the Base64 converted certificate (.pem) file to the **trusted.cer** file.

```
cat <certificate name>.pem >>
opt/protegility/alliance/config/outbound_certs/trusted.cer
```

Note: If you started with a different certificate file extension name that was already ASCII Base64 encoded, then use the same extension name instead of the “.pem” file extension name in the command example above.

8. In the ESA Web UI, navigate to **Cloud Gateway > RuleSet**.
9. Select the service to which the certificate is assigned and click **Edit**.
10. Add the following option in the **Outbound Transport Settings** textbox.

```
{"ca_certs": "/opt/protegility/alliance/config/outbound_certs/trusted.cer"}
```

11. Restart the **Cluster** by navigating to **Cloud Gateway > Cluster** to replicate certificates to all DSG nodes.

12.2.2 Use Learn Mode to Find Message Carrying Sensitive Data

When enabled, Learn mode logs all message exchange to host names matching the service configuration. This allows us to consider these messages payload, learn how it is structured so appropriate rules can be set to transform the relevant parts in it before it is forwarded on.

Let's first check that Learn mode is functioning properly by verifying that new entries are shown as the **www.ffcrm.biloxi.com** application is accessed. Navigate the main menu **Cloud Gateway -> RuleSet -> Learn Mode**. If this is the first time you're

visiting the Learn mode page then chances are it will come up with no data. You can always click the reset () button to purge the learn mode data.

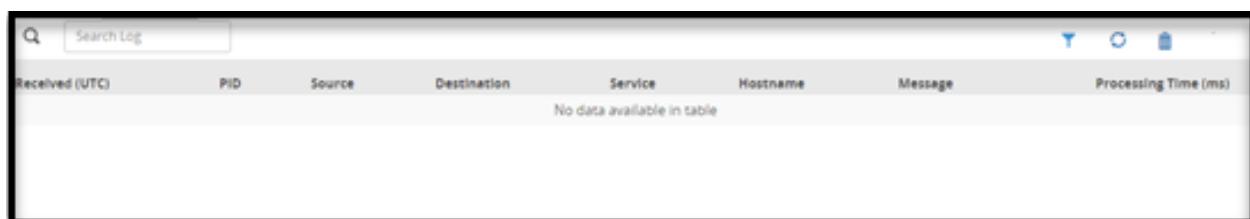


Figure 12-4: Learn Mode screen

Navigating to the **www.ffcrm.biloxi.com** application home page will generate new entries in the Learn mode page. It is recommended to use a separate window so that going back to the Learn mode page becomes easier.

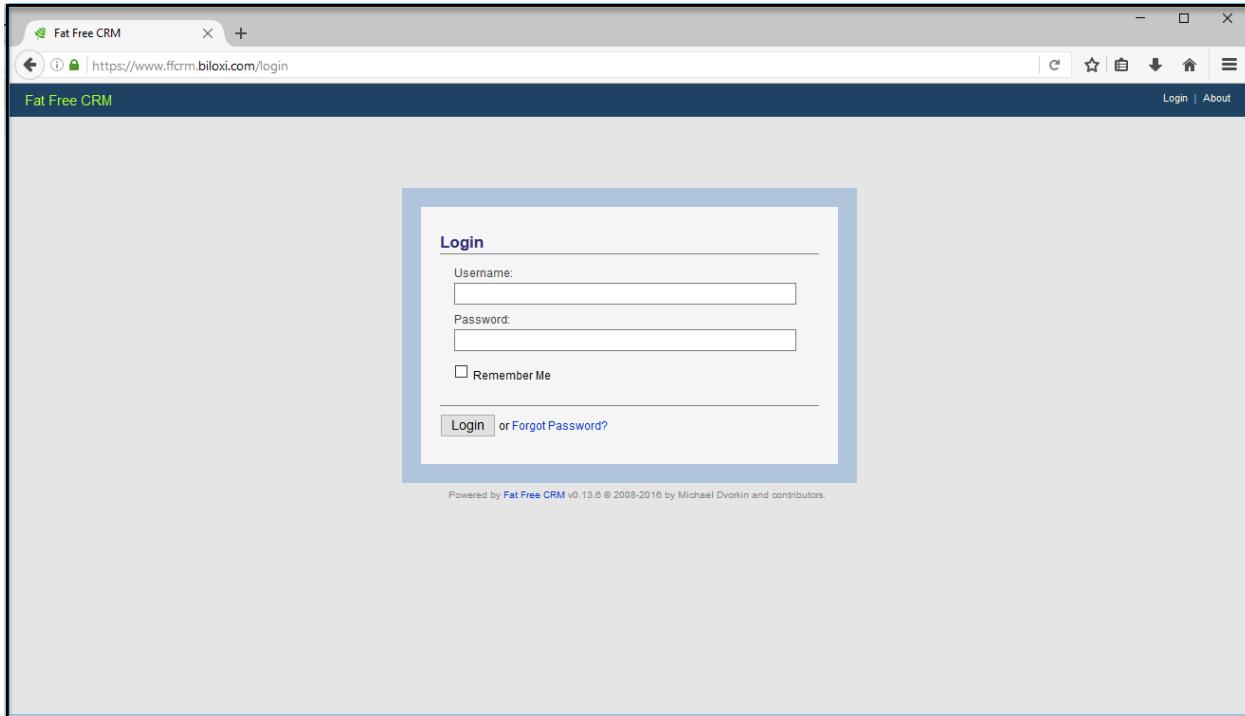


Figure 12-5: CRM login screen

Switch back to the Learn page and refresh the list by clicking the refresh () button. New entries will indicate that Learn mode is functioning as expected.

< Cloud Gateway		Monitoring	Learn Mode	Log Viewer					
Cluster									
RuleSet									
Transport									
			Search Log						
				Processing Time (ms)					
		Received (UTC)	PID	Source	Destination	Service	Hostname	Message	
		2016-04-28 22:21:13.965297	8706	www.ffcrm.com	10.10.10.191	ffcrm.biloxi.com	node02	200 OK (2016-04-28 22:21:13.847606 UTC)	14.279
		2016-04-28 22:21:13.847606	8706	10.10.10.191	www.ffcrm.com	ffcrm.biloxi.com	node02	GET https://www.ffcrm.biloxi.com/assets/application-89ab63927831aed2b99a94d4128d1...	0.932
		2016-04-28 22:21:13.799137	26085	www.ffcrm.com	10.10.10.191	ffcrm.biloxi.com	node01	200 OK (2016-04-28 22:21:13.748156 UTC)	1.726
		2016-04-28 22:21:13.748156	26085	10.10.10.191	www.ffcrm.com	ffcrm.biloxi.com	node01	GET https://www.ffcrm.biloxi.com/login	1.615

Figure 12-6: Learn Mode screen

12.2.2.1 Password Masking

The **www.fferm.biloxi.com** application requires users to authenticate with their username and password before they can use the application. To avoid Learn mode from logging user's passwords, Service' Password Masking needs to be configured. The Password Masking configuration comprised of a regex pattern to identify the URL(s), another regex pattern to identify the password in the payload and a masking value which will be replace the password before the message is logged by Learn mode.

To do that, the request message carrying the password needs to be identified when the authentication is submitted. Using a unique value for the password will make it easier to find in Learn mode. No need to use real username and password, instead let's use testusertest for the username and testpasswordtest for the password.

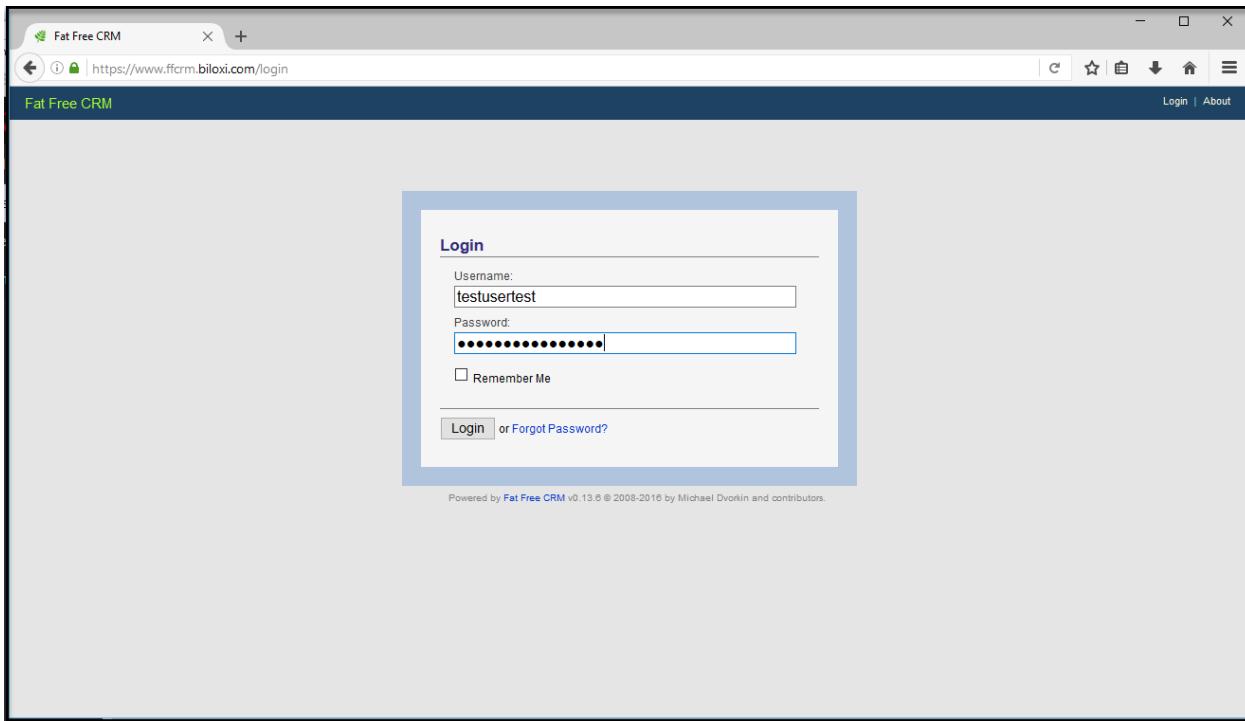


Figure 12-7: CRM login screen

Submit the login form, switch to the *Learn mode* screen and refresh it.

Received (UTC)	PID	Source	Destination	Service	Hostname	Message	Processing Time (ms)
2016-04-28 22:22:42.676620	8707	www.ffcrm.com	10.10.10.191	ffcrm.biloxi.com	node02	200 OK (2016-04-28 22:22:42.535278 UTC)	17.713
2016-04-28 22:22:42.535278	8707	10.10.10.191	www.ffcrm.com	ffcrm.biloxi.com	node02	GET https://www.ffcrm.biloxi.com/assets/application-89ab63927831aed2b99a94d412bd1...	4.175
2016-04-28 22:22:42.481306	26085	www.ffcrm.com	10.10.10.191	ffcrm.biloxi.com	node01	200 OK (2016-04-28 22:22:42.344539 UTC)	2.684
2016-04-28 22:22:42.244529	26085	10.10.10.191	www.ffcrm.com	ffcrm.biloxi.com	node01	GET https://www.ffcrm.biloxi.com/login	1.479
2016-04-28 22:22:42.235095	8706	www.ffcrm.com	10.10.10.191	ffcrm.biloxi.com	node02	302 Found (2016-04-28 22:22:42.175395 UTC)	0.936
2016-04-28 22:22:42.175395	8706	10.10.10.191	www.ffcrm.com	ffcrm.biloxi.com	node02	POST https://www.ffcrm.biloxi.com/authentication	1.249
2016-04-28 22:21:13.965297	8706	www.ffcrm.com	10.10.10.191	ffcrm.biloxi.com	node02	200 OK (2016-04-28 22:21:13.847606 UTC)	14.279
2016-04-28 22:21:13.847606	8706	10.10.10.191	www.ffcrm.com	ffcrm.biloxi.com	node02	GET https://www.ffcrm.biloxi.com/assets/application-89ab63927831aed2b99a94d412bd1...	0.932
2016-04-28 22:21:13.799137	26085	www.ffcrm.com	10.10.10.191	ffcrm.biloxi.com	node01	200 OK (2016-04-28 22:21:13.748156 UTC)	1.726
2016-04-28 22:21:13.748156	26085	10.10.10.191	www.ffcrm.com	ffcrm.biloxi.com	node01	GET https://www.ffcrm.biloxi.com/login	1.615

Figure 12-8: Learn mode after logging to CRM screen

Additional messages now appear in the list. Let's examine the first entry listed right after submitting the fake credentials by clicking on it. Additional details will appear at the bottom of the page. Click any entry on the left, select the *Messages Difference* tab on the right to see the message payload and scroll all the way down.

The screenshot shows the 'Learn Mode' analysis interface. At the top, there's a table of logs with columns: Received (UTC), PID, Source, Destination, Service, Hostname, Message, and Processing Time (ms). Below the table is a detailed view of a selected log entry. This view includes tabs for 'Filter Summary', 'Message Difference', and 'Wrap lines'. The 'Message Difference' tab displays the raw message content, which is heavily redacted with blue circles. The redacted content includes a URL (POST https://www.ffcrm.biloxi.com/authentication) and a large amount of POST data. The bottom right of the interface shows a status bar with 'Last Refreshed: Thu, Apr 28 2016 12:46 PM' and a refresh button.

Figure 12-9: Learn Mode analysis screen

Note: The rules displayed in the *Learn mode* screen and the payload message difference are stored in a separate file in DSG. The default file size limit for this file is 3MB. If the payloads and rules in your configuration are high in volume, you can configure this file size.

In the ESA CLI Manager, navigate to **Administration > OS Console**. Edit the `webuiConfig.json` file in the `/opt/protegility/alliance/config/webinterface` directory to add the Learn mode file size parameter as follows:

```
{
  "learnMode": {
    "MAX_WORKER_THREADS": 5,
    "LEARNSIZE_FLOW_DUMP_FILESIZE": 10
  }
}
```

You can now see that the characteristics of the message carrying the authentication password can be the request message Method and URL. It is evident that the authentication with **www.ffcrm.com** is handled by a **POST** request to **/authentication** URI and that the password is carried in this message body as a value for the parameter name **authentication%5Bpassword%5D** (which is URI/www-form encoding for **authentication[password]**).

Note that in some cases, you may need to repeat this test several times to confirm the consistency of the URL, parameter names, etc.

With these details, you can adjust the service configuration with the details that were found to mask the authentication password (or any other data is not required by learn mode to log in the clear). Both the pattern and the resource are regular expressions so let's use `/authentication$` for the resource and `(?<=(\bauthentication%5Bpassword%5D=))(.+?)(?=$|&)` for the pattern.

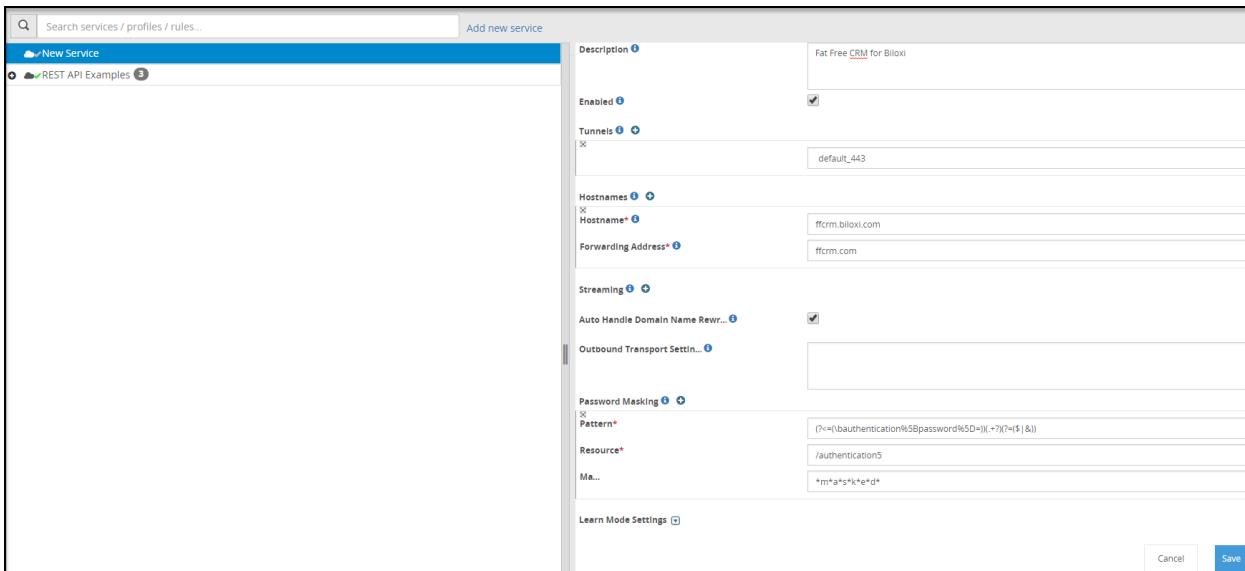


Figure 12-10: Create Rules screen

Save the changes and restart the cluster for these changes to take effect. You can do that from the **Cloud Gateway -> Cluster** page, **Monitoring** tab.

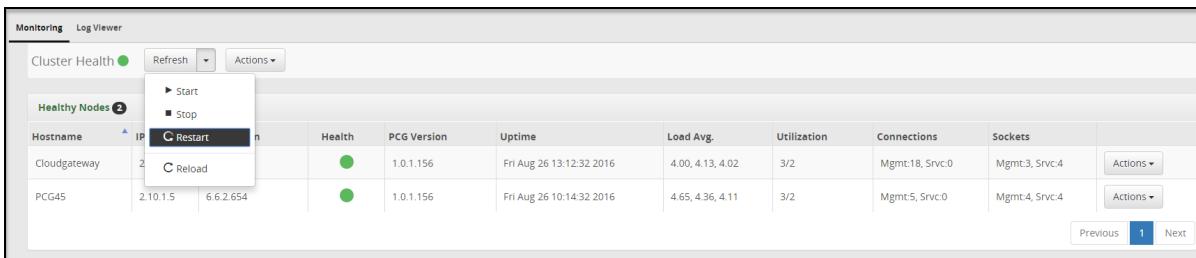


Figure 12-11: Restart Cluster screen

Test the change made to the service settings by repeating the authentication attempt as done before. Again, no need to use real credentials, as this step is to ensure the password gets masked in Learn mode.

Figure 12-12: Learn Mode analysis screen

Note that resetting the Learn mode list makes it easier to find the messages you are looking for in a smaller set of message.

12.2.2.2 Protecting Sensitive Data

Data protection is achieved by finding the target data in a message and transforming it from its clear state to protect and back depending on the flow direction of the message. In the **www.ffcrm.com** application the account names need to be protected, therefore the gateway will be configured to protect account names in messages transmitting it from user's browser to the application backend. The gateway will also be configured to remove the protection (unprotect) when it is consumed by transforming it in message payloads transmitted from the application backend back to user's browser.

Following the Protegility methodology, the discovery phase will uncover all the sensitive data entry points in our target application. One obvious entry point is the page in the application where users can create new accounts. Navigate **www.ffcrm.biloxi.com** to where new accounts and purge the Learn mode list before creating a new testing account. A sample TestAccount123 is unique enough name to easily find in Learn mode.

Figure 12-13: Create Account in CRM screen

Switching back to Learn mode page, the message posed with the account name TestAccount123 that was created can be found.

The screenshot shows the 'Learn Mode' tab selected in the top navigation bar. The main area displays a log entry for a POST request to `/accounts`. The message body is shown in URL-encoded format, containing the account name `TestAccount123`.

```

Received (UTC) PID Source Destination Service Hostname Message Processing Time (ms)
2016-04-29 19:20:52.218297 52889 www.ffcrm.com 10.10.10.191 ffcrm.biloxi.com node01 200 OK (2016-04-29 19:20:51.916595 UTC) 1.996
2016-04-29 19:20:51.916595 52889 10.10.10.191 www.ffcrm.com ffcrm.biloxi.com node01 POST https://www.ffcrm.biloxi.com/accounts 1.877

Flow Host:node01 Tunnel:default_443 Service:ffcrm.biloxi.com
Showing All rules
Filter Summary Message Difference Wrap lines
2c726b19417cb780f4ec1fb0d861e7ddzeed267dd94b1ec263408873ee
65ad883d14a19784215332%3A%3A9
14 Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-CSRF-Token: CBvR0uRo3Z5RmuJptTxgAvnHR|jGIClWigTU4-
16
17 utr8=%E2%9C%93&authenticity_token=CBvR0uRo3Z5RmuJptTxgAvnH
jGIClWigTU4-3D&account%5Buser_id%5D=98account%5Bname%5D=
TestAccount123&account%5Bassigned_to%5D=&account%5Bcate
gory%5D=&account%5Btag%5D=0&account%5Btag_list%5D=&comm
ent_body=&account%5Btoll_free_phone%5D=&account%5Bphone%5D=
&account%5Bfax%5D=&account%5Bwebsite%5D=&account%5Bemail%5D=
&account%5Bbilling_address_attributes%5D%5Baddress_type%5D=
Billing&account%5Bbilling_address_attributes%5D%5Bstreet1%5D=&c
ount%5Bbilling_address_attributes%5D%5Bstreet2%5D=&account%5B
billing_address_attributes%5D%5Bcity%5D=&account%5BBilling_addres
s_attributes%5D%5Bstate%5D=&account%5BBilling_address_attributes%
5D%5Bzip_code%5D=&account%5BBilling_address_attributes%5D%5Bc
ountry%5D=&account%5Bshipping_address_attributes%5D%5Baddress_
type%5D=&account%5Bshipping_address_attributes%5D%5Bstreet1%5D=
&account%5Bshipping_address_attributes%5D%5Bstreet2%5D=&account%5B
shipping_address_attributes%5D%5Bcity%5D=&account%5Bshippin
g_address_attributes%5D%5Bstate%5D=&account%5Bshipping_address_attributes%
5D%5Bzip_code%5D=&account%5Bshipping_address_attributes%5D%5Bc
ountry%5D=&account%5Bshipping_address_attributes%5D%5Bshippin
g_address_attributes%5D%5Bstreet1%5D=&account%5Bshipping_address_attributes%
5D%5Bcity%5D=&account%5Bshipping_address_attributes%5D%5Bstate%5D=&acc
ount%5Bzip_code%5D=&account%5Bgroup_ids%5D%5Bgroup_id%5D=&acc
ount%5Bgroup_name%5D=&commit=Create+Account

```

Figure 12-14: Learn mode analysis screen

Learn mode is showing that DSG sees this information sent as a value to the **account[name]** parameter in the **POST** request message body to **/accounts** URI. The message body is encoded using URL Encoding which is again why **account[name]** is displayed as **account%5Bname%5D**.

You have enough details now to create a new set of rules under Ruleset to extract the posted account name and protect it using Protegility Data Protection transformation. Create a profile under the existing service and call it Account Name.

The screenshot shows the 'Create Rules' screen. A new rule is being created for the `ffcrm.biloxi.com` service, named `Account Name`. The rule is currently active and enabled. The description is `Rules for protecting account names`.

Figure 12-15: Create Rules screen

Now create a new rule which will extract the body of the create account message. This rule will look for a **POST HTTP Request** message with URI matching the regular expression **/accounts\$**.



Name*: Create Account Message
Descript...: Extract the body of the create account message.
Enabl...:
Action*: Extract
Payload*: HTTP Message
HTTP Message Type*: HTTP Message
HTTP Method: POST
URI: /account\$
Request Headers:
Message Body:
Require Client Certificate...:
Authentication: None
Target Object: Message Body

Figure 12-16: Create Rule screen

Under the Create Account Message rule, let's add another rule to extract the value of the parameter which carries the account name found earlier that this parameter name is **account[name]**.

Name*: Account Name Parameter
Descript...: Extract the value of the account name parameter.
Enabl...:
Action*: Extract
Payload*: HTML Form Media Type (X-WWW-FORM-URL_ENCODED)
Name: account[name]
Value: Value
Target Object: Value
Encoding Mode: Automatic (Favoring Standard Encoding)
Encoding Reserve Character..: -_!^{}()

Figure 12-17: Create Rule screen

Last the account name needs to be protected using a Protegility Data Protection Transformation rule. This rule requires us to specify and configure how to protect the data. Data Element is a value defined in data security policies managed by Security Officers using Protegility Enterprise Security Administrator (ESA), which represents data security characteristics like encryption or tokenization. Our security officer had provided us with a data element name **PTY_ACCOUNT_NAME** which was created for protecting account names in Biloxi corp.

Note:

The **alliance** user is an OS user and is responsible for managing DSG processes. Ensure that you do not add this user as a policy user.

For more information about other OS users, refer to the section *OS Users in Appliances* in the [Protegility Appliances Overview Guide 9.1.0.5](#).

For more information about the Protegility Data Security policy, refer to [Protegility Policy Management Guide 9.1.0.5](#).

In addition to the data protection, an encoding codec with unique prefix and suffix will be used. The reason for it is so that the protected data will be easy to find by looking for data which begins and ends with the specified prefix and suffix. The ability to find it easily comes handy when the data is consumed and a set of rules must be created to remove the protection. Instead of creating a rule for every message that might carry it, we can scan every message coming back from www.ffcrm.biloxi.com and unprotect every protected data we find corresponding to our prefix and suffix.

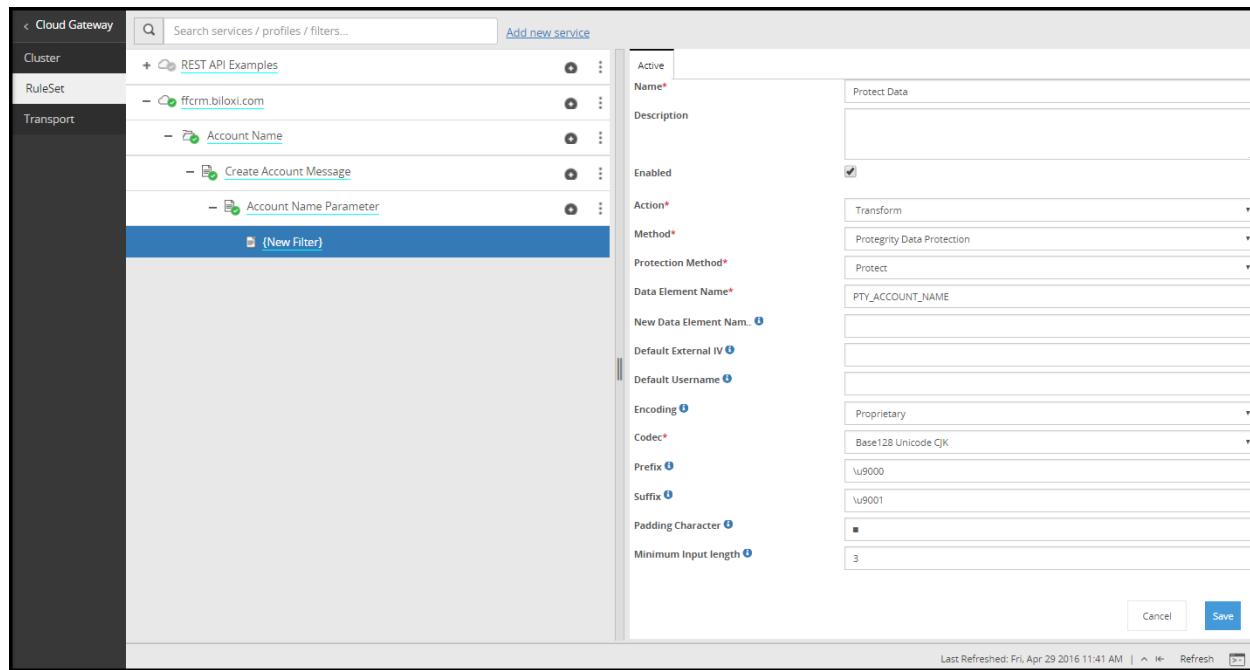


Figure 12-18: Create Rule screen

Restart the gateway cluster and test the new set of rules that was created by creating a new account name **TestAccount456**.

The expected result is that the new account name will appear protected (encrypted/tokenized and encoded) as oppose to the same way it was entered.

The screenshot shows the 'Accounts' page of the Fat Free CRM application. On the left, there's a sidebar with 'Account Categories' (Affiliate, Competitor, Customer, Partner, Reseller, Vendor, Other) and 'Global lists' (No saved lists). Below that is a 'Recent items' section with a list of accounts. One item in this list is circled in red: 'Account: 退擋套機狗被淫導燒英 艾達' (Account: TestAccount123). The main content area shows a list of 121 accounts, each with a category color-coded box (e.g., Affiliate, Competitor, Customer, Partner, Reseller, Vendor), the account name, and a timestamp. The first account listed is 'Other 退擋套機狗被淫導燒英 艾達 - added 3 days ago by Administrator | 0 contacts | 0 opportunities'.

Figure 12-19: View protected text screen

Last the data protection applied on Account Names needs to be reversed so that authorized Biloxi users consuming this information through www.ffcrm.biloxi.com will see it in clear.

The prefix and suffix will help searching the protected account names in every payload coming back from the application. Therefore a generic rule for textual payloads can be created, like HTML which will look for protected account names and unprotect them such that users will get the HTML payload with the account name in the clear.

The top level branch in this case will target a HTML Response message.

Search services / profiles / filters... Add new service

Active

Name* Account Name HTML

Description Extract HTML messages carrying protected account names

Enabled

Action* Extract

Payload* HTTP Message

HTTP Message Type* Response

Method

Request URI

Request Headers

Request Body

Response Status Code

Response Headers

- Name*** Content-Type
- Value*** html

Response Body

Target Object Message Body

Cancel Save

Last Refreshed: Mon, May 2, 2016 10:22 AM | Refresh

Figure 12-20: Create Rule screen

In the extracted HTML Response message body, we will look for the protected account names by searching for values matching a regex pattern comprised of our prefix and suffix.

Cloud Gateway Search services / profiles / filters... Add new service

Cluster

RuleSet

Transport

Active

Name* Protected Account Names

Description Extract protected account names in HTML encoded payload

Enabled

Action* Extract

Payload* Text

Prerequisite Match Pat...

Pattern

Pattern Group Id 0

Encoding External

Codec* HTML Encoding

Cancel Save

Last Refreshed: Mon, May 2, 2016 10:22 AM | Refresh

Figure 12-21: Create Rule screen

The leaf rule of this branch will unprotect the extracted protected account names found in the HTML response payload.

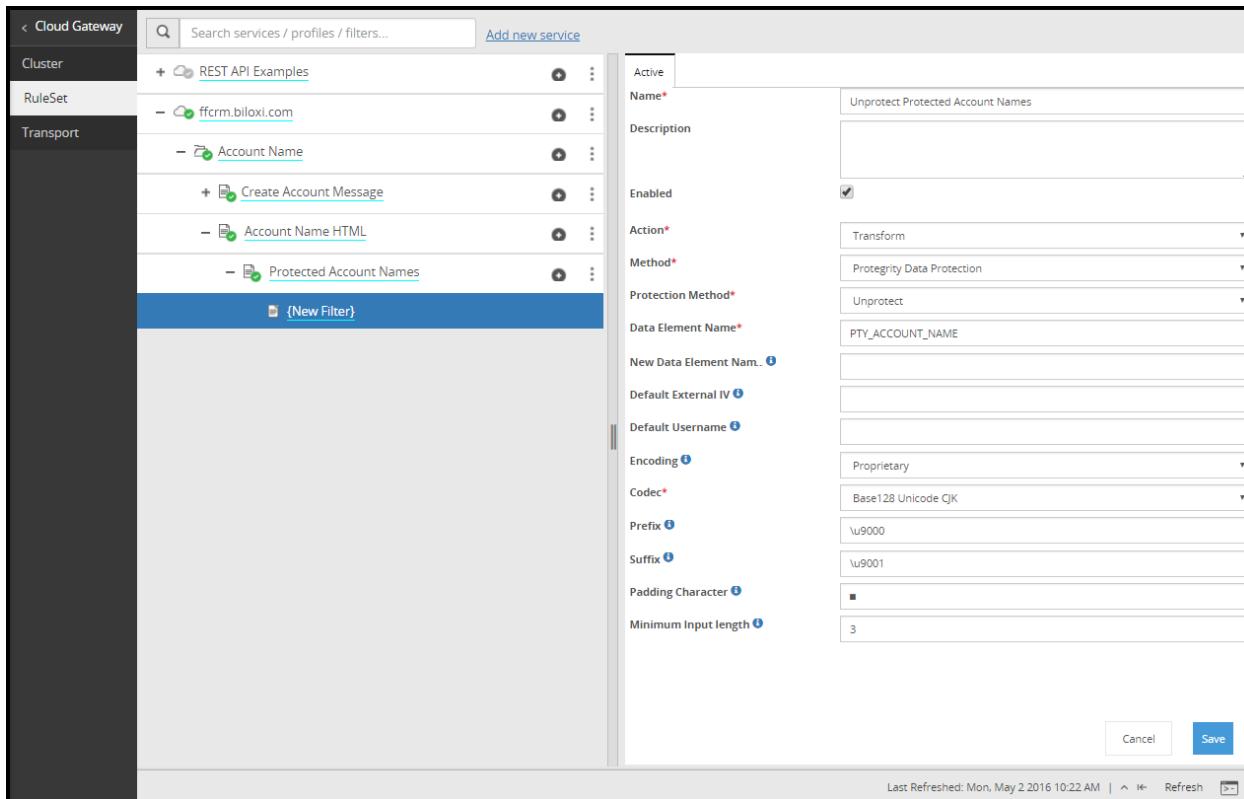


Figure 12-22: Create Rule screen

Let's restart the gateway cluster and test the new rules branch created by revisiting the page where **TestAccount456** account name appeared protected earlier.

Note: A simple refresh of the web page may not be enough as local caching may be used by the browser. Clearing cache, local storage and session information may be required.

Figure 12-23: View unprotected text screen

Other payload types such as JSON or XML may be used by the application therefore additional generic unprotect rules for these payload types may be required.

12.3 Forwarding logs to SIEM systems

If you plan to forward all ESA and gateway logs to an external Security Information and Event Management (SIEM) system, such as Splunk or AWS Cloudwatch, you must refer this section.

12.3.1 Forwarding logs to an external SIEM

If you plan to forward all ESA and gateway logs to an external Security Information and Event Management (SIEM) system, you must configure the `alliance.conf` file.

► To forward logs to an SIEM system:

1. In the ESA Web UI, navigate to **Settings > System**.
2. Navigate to the **Cloud Gateway > Settings** area.
3. Click **Edit** to edit the `alliance.conf` file.
4. You can either send all logs to the SIEM system or just the gateway logs.
 - a. To send gateway or DSG logs using TCP, add the following rule to the file.

```
*.* @@(<IP_ADDRESS> or <HOSTNAME>):<PORT>
```

- b. To send gateway or DSG logs using TCP, add the following rule to the file.

```
:msg, contains, "PCPG:" @@(<IP_ADDRESS> or <HOSTNAME>):<PORT>
```

Note:

- Ensure that the rule is the first entry in the `alliance.conf` file.
 - The configurations must be made in ESA. After you deploy the configuration, ensure that the rsyslog service is restarted on each DSG node.
 - If you are using UDP protocol instead of TCP protocol, the rules to be defined are as follows:
1. To send gateway or DSG logs using UDP, add the following rule to the file.

```
*.* @(<IP_ADDRESS> or <HOSTNAME>):<PORT>
```

2. To send gateway or DSG logs using UDP, add the following rule to the file.

```
:msg, contains, "PCPG:" @(<IP_ADDRESS> or <HOSTNAME>):<PORT>
```

The following file is a sample `alliance.conf` file that shows the SIEM configurations.

```

#/etc/rsyslog.d/alliance.conf Configuration file for cloud gateway syslog
#
#To forward syslog events to an external system, add a rule at the beginning
#with following syntax
#1. For sending all the logs - . @@(IP/HOSTNAME):PORT
#2. For sending only gateway logs - :msg, contains, "PCPG:" @@(IP/HOSTNAME):PORT
#To transfer the logs using UDP instead of TCP, replace @@ with @
#
#FOR THE CHANGES MADE HERE TO TAKE EFFECT, RSYSLOG SERVICE MUST BE RESTARTED
ON EACH DSG NODE IN CLUSTER
#
#Below are couple of examples, uncomment them and replace ip and port with your
# remote server's ip/hostname and port:
#. @@192.168.0.1:514
#:msg, contains, "PCPG:" @@192.168.0.1:514

:msg, contains, "reportTransactionMetrics" @10.0.0.65:514

#Filter the incoming msg here .If the Meassage has "PCPG:" string then
#forward the msg to /var/log/gateway file.
:msg, contains, "PCPG:" /var/log/gateway.log
:app-name, contains, "BDLP:" /var/log/gateway.log

#if the message contains PCPG: string then wont write into /var/log/messages
# & /var/log/syslog
:msg, contains, "PCPG:" ~ /var/log/messages
:msg, contains, "PCPG:" ~ /var/log/syslog
:app-name, contains, "BDLP:" ~ /var/log/messages
:app-name, contains, "BDLP:" ~ /var/log/syslog

```

Figure 12-24: Alliance.conf file configurations

12.3.2 Forwarding logs to AWS CloudWatch

If you plan to forward logs from an DSG appliance to AWS CloudWatch, then you must refer to this section.

The AWS CloudWatch must be configured with each DSG node in the cluster to forward the logs from an DSG appliance.

For more information about how to configure AWS CloudWatch with an appliance, refer to section *Working with Cloud-based Applications* in the [Protegility Appliances Overview Guide 9.1.0.5](#).

After the AWS CloudWatch is configured with a DSG node, you must add the path of the required DSG appliance log files to the *appliance.conf* file in the appliance.

For more information about adding custom logs to the *appliance.conf* file and forwarding them to AWS CloudWatch, refer to section *Configuring Custom Logs on AWS CloudWatch Console* in the [Protegility Appliances Overview Guide 9.1.0.5](#).

12.4 Rolling Out to Production

Following the Protegility methodology, production rollout requires deploying DSG with all the necessary rules to protect and unprotect sensitive data. You also need to prevent users/applications bypassing DSG to avoid sensitive data in the clear from reaching the target application directly. Lastly, migrate any historical data that may already exist in the target system.

After Ruleset is created and tested in sandbox or testing environment you can backup and export the entire configuration and then import and restore it on the production DSG cluster. Addresses, certificates and data elements names may be different from a non-production system which may require some modification to it which applies to production environment only. Before going live it is recommended to test and verify the Ruleset configuration for all sensitive data in scope of the workflow mapped during discovery phase. Beyond and after a successful testing of the workflows, one should also temporarily be bypassing DSG to confirm that all sensitive data resided protected on the application backend side and that no backend instance of the target sensitive data exists in the clear.

Target application may still be accessible directly, especially when the target application is a public SaaS. For example, after production rollout, Biloxi users may still attempt to access www.ffcrm.com directly. The risk is not in these users seeing protected data as much as it is in them submitting data which otherwise would have been protected by DSG before reaching the application backend. There are several ways to prevent that and may not even be a challenging factor when the target application

is owned by the same organization who implements DSG. If however the application is external, i.e. SaaS, controlling the authentication can be used to solve this problem.

Authentication may be offered by SaaS using SAML or by owning the authentication process themselves. Most organization would prefer Single Sign On (SSO) using SAML which DSG can be configured to proxy or resign, essentially establishing a trust between DSG and the SaaS. In cases however when SAML is not an available option, DSG can be configured to treat username/password as sensitive data as well. This means that users will be known to the SaaS as their protected username and password. An attempt to bypass DSG in such case will result the same as it would for users who are not known to the application at all.

It is not uncommon to introduce data protection to a system already in use. In such cases, it is likely that data designated for protection has already been added to the system in the clear. Historical data migration is the process of replacing the sensitive data from its clear state to a protected state. Applications may offer different ways of achieving this goal normally by exporting the data, transforming it and importing it back into the system. DSG REST API can be used to perform the transformation which will require a Ruleset to be configured for the exported payload format.

Homegrown and third party application changes all the time and therefore it is highly recommended to maintain a testing environment for future Ruleset or other configuration modification that may be needed.

Chapter 13

Transaction Metrics Logging

13.1 Total Time Breakdown for HTTP Request

The transaction metrics allows the user to view the detailed information of the operations performed by the DSG. The Transaction metrics logging feature can be enabled at the service level.

For more information about enabling the transaction metrics logging feature, refer to the [Table: Service Fields](#).

The transaction metrics are logged in the `gateway.log` file in JSON format.

The sample transaction metrics for an HTTP request is as seen in the following snippet.

```
{
  "auth_cache_hit":false,
  "auth_end_time": "2023-06-01T10:00:00Z",
  "auth_start_time": "2023-06-01T09:59:55Z",
  "auth_total_time": 0.013525,
  "auth_user_name": "admin",
  "client_ip": "192.168.1.123",
  "end_time": "2023-06-01T10:00:00Z",
  "dsg_version": "3.1.0.5",
  "http_method": "POST",
  "http_outbound_available_clients": 100,
  "http_outbound_count_new_connections": 1,
  "http_outbound_count_redirect": 0,
  "http_outbound_local_port": 57494,
  "http_outbound_remote_ip": "192.168.1.123",
  "http_outbound_size_download": 15.00,
  "http_outbound_size_queue": 0,
  "http_outbound_size_upload": 15.00,
  "http_outbound_speed_download": 5305.00,
  "http_outbound_speed_upload": 5305.00,
  "http_outbound_time_appeared": 0.000000,
  "http_outbound_time_connect": 0.000974,
  "http_outbound_time_namelookup": 0.000378,
  "http_outbound_time_pretransfer": 0.001021,
  "http_outbound_time_queue": 0.000000,
  "http_outbound_time_redirect": 0.000000,
  "http_outbound_time_request": 0.003088,
  "http_outbound_time_starttransfer": 0.001025,
  "http_outbound_time_total": 0.002827,
  "http_outbound_url": "http://sample.server.com:8081/echo/basicAuth",
  "http_outbound_status_code": "OK",
  "http_status_code": 200,
  "learn_mode_enabled": false,
  "local_port": 44296,
  "node_hostname": "node-1",
  "node_id": 15314,
  "open_connections": 1,
  "pre_processing_time": 0.000793,
  "processing_time_downstream": 0.013261,
  "processing_time_request": 0.03502,
  "processing_time_response": 0.001667,
  "request_url": "http://sample.protected-server7.com:8081/echo/basicAuth",
  "service_name": "HTTP Service Tests",
  "service_type": "HTTP-GW",
  "start_time": "2023-06-01T10:00:00Z",
  "total_time": 0.026503,
  "transformations": 1,
  "transformations_rules": [
    {
      "regex_replace": [
        {
          "replace_rules": [
            {
              "rule_name": "Replace Sensitive Data",
              "num_replace": 2
            }
          ]
        }
      ]
    }
  ],
  "tunnel_name": "auto_8081",
  "user_name": "admin"
}
```

Figure 13-1: Sample transaction metrics

The following table describes the parameters available in the transaction metrics for different services.

Table 13-1: Transaction metrics parameters

Parameter	Supported on Service	Data type	Description	Additional Information
auth_cache_hit	HTTP, REST	boolean	True indicates that the basic authentication credentials were cached and False	

Parameter	Supported on Service	Data type	Description	Additional Information
			indicates that the credentials were not cached.	
auth_start_time	HTTP, REST	string	Timestamp when the basic authentication was started.	
auth_end_time	HTTP, REST	string	Timestamp when the basic authentication was completed.	
auth_total_time	HTTP, REST	float	The difference in seconds between the <i>auth_end_time</i> and <i>auth_start_time</i> parameters.	
auth_user_name	HTTP, REST	string	Username used for basic authentication.	
bucket_name	S3 Out-of-Band	string	The name of the S3 bucket from where the DSG reads the object to be processed.	
client_correlation_handle	All	string	The ID used to uniquely identify a request. It is usually a UUID.	This parameter is optional.
client_ip	All	string	The IP address of the client that sent the request to the DSG.	
dsg_version	All	string	The version number of the gateway process.	
file_name	S3 Out-of-Band, Mounted Out-of-Band	string	The name of the file that has been processed.	
http_method	HTTP, REST	string	The HTTP method associated with request.	
http_outbound_available_clients	HTTP	integer	The number of outbound HTTP clients available for the requests.	
http_outbound_count_new_connections	HTTP	integer	The number of new connections created to process the request.	A single request per client can be processed. Whenever a connection is created, it is cached and it can be reused. If an existing connection is used, then the value of this parameter is 0. If new connections are created, then the value of this parameter will be greater than 0.
http_outbound_count_redirect	HTTP	integer	The number of redirects encountered while processing a request.	
http_outbound_local_port	HTTP	integer	The local port used for the outbound connection.	
http_outbound_response_code	HTTP	integer	The HTTP status response code from downstream system.	

Parameter	Supported on Service	Data type	Description	Additional Information
http_outbound_size_download	HTTP	float	The size of the data received from the downstream system in bytes.	
http_outbound_size_queue	HTTP	float	The number of requests waiting to be sent to downstream systems.	
http_outbound_size_upload	HTTP	float	The size of data sent to downstream system in bytes.	
http_outbound_speed_download	HTTP	float	Average download speed. Bytes per second.	
http_outbound_speed_upload	HTTP	float	Average upload speed. Bytes per second.	
http_outbound_time_appconnect	HTTP	float	The time taken to complete the SSH/TLS handshake.	
http_outbound_time_connect	HTTP	float	The time taken to connect to the remote host.	
http_outbound_time_namelookup	HTTP	float	The time taken to resolve the name.	
http_outbound_time_pretransfer	HTTP	float	The time from the start until before the first byte is sent.	
http_outbound_time_queue	HTTP	float	The time that the requests spent in the queue before being processed.	
http_outbound_time_request	HTTP	float	The time from when the request was popped off the queue to be processed to the time a response was sent back to the caller.	
http_outbound_time_starttransfer	HTTP	float	The time taken from the start of the request until the first byte was received from the server.	
http_outbound_time_total	HTTP	float	Total time that the client library took to process the HTTP request.	
http_outbound_url	HTTP	string	The destination URL used for the outbound request.	
http_outbound_time_redirect	HTTP	float	The time, in seconds, it took for all redirection steps including name lookup, connect, pretransfer, and transfer before the final transaction was started. The <i>http_outbound_time_redirect</i> shows the complete execution time for multiple redirections.	
http_reason_phrase	HTTP, REST	string	The reason phrase associated with HTTP status code.	
http_status_code	HTTP, REST	integer	The HTTP status code sent to HTTP client.	
input_etag	S3 Out-of-Band	string	The Etag of the input object processed by the DSG.	
input_size	S3 Out-of-Band	integer	The size of the input object, in bytes, processed by the DSG.	

Parameter	Supported on Service	Data type	Description	Additional Information
learn_mode_enabled	All	boolean	Indicates if the learn mode is enabled.	
local_port	HTTP, REST	integer	The local port used for the inbound connection, can be used with the <i>open_connections</i> parameter to identify new and unique connections.	
method	SFTP	string	The SFTP method associated with the request. The method can be either GET or PUT.	
node_hostname	All	string	The hostname of the DSG.	
node_pid	All	integer	The process id of the gateway process that processed the request.	
open_connections	HTTP, REST	integer	The number of open connections associated with the tunnel in a process.	
output_bucket_name	S3 Out-of-Band	string	The name of S3 bucket where the DSG writes the processed object.	
output_etag	S3 Out-of-Band	string	Etag of the output object processed by the DSG.	
output_file_name	S3 Out-of-Band	string	The name of the object that is written to the new S3 bucket (i.e. The value of <i>output_bucket_name</i> parameter) by the DSG.	
output_size	S3 Out-of-Band	integer	The size of the object, in bytes, written to the output S3 bucket.	
time_lock	S3 Out-of-Band	float	The time taken to process the file from the time the lock was created.	
unlock_time	S3 Out-of-Band	float	The time taken for a lock object associated with an S3 object (file) to get unlocked or deleted successfully.	
processing_time_downstream	HTTP, SMTP, SFTP	float	The time is the difference between the start time of processing a response and the end time of processing a request.	
processing_time_request	All	float	The time taken for the ruleset to process the request data.	
processing_time_response	HTTP, SMTP, SFTP, S3 Out-of-Band	float	The time taken for the ruleset to process the response data. It is only applicable to the protocols where a response is expected from a downstream system.	
request_uri	HTTP, REST	string	The URI of the request being processed by the DSG.	
service_name	All	string	The name of the service processing the request.	

Parameter	Supported on Service	Data type	Description	Additional Information
service_type	All	string	The type of the service processing the request.	
server_ip	SFTP	string	The IP address of the SFTP server that the DSG is communicating with.	
end_time	All	string	The timestamp representing when a request was completed.	
pre_processing_time	HTTP, REST	float	The time an HTTP or REST request waited before it was processed.	
start_time	All	string	The timestamp when the DSG received a request.	
total_time	All	float	If the <i>pre_processing_time</i> is 0, then the <i>total_time</i> is the difference between the <i>end_time</i> and <i>start_time</i> parameters. If the <i>pre_processing_time</i> is greater than 0, then the <i>total_time</i> is the difference between the <i>end_time</i> and <i>pre_processing_time</i> parameters.	
fd	HTTP, REST	integer	The number of open file descriptors.	
tunnel_name	All	string	The name of the tunnel processing the request.	
transformations	All	object	The object representing the Regex Replace and Protegility Data Protection transformation rules.	
data_protection	All	object	The object representing the Protegility Data Protection transformation rule.	
regex_replace	All	object	The object representing the Regex Replace transformation rule.	
user_name	All	string	The username used for the protection, unprotection, or reprotection.	
data_element_name	All	string	The name of the data element used to transform the sensitive data.	
in_payload_len	HTTP, REST	unsigned long long integer	The length of the input payload.	
len_protect	All	integer	The length of the sensitive data that is protected.	
num_protect	All	integer	The number of protect operations performed.	
rule_name	All	string	The name of the rule used to transform the sensitive data.	
num_replace	All	integer	The number of regex replace performed.	

In the [Figure: Sample transaction metrics](#), some of the timestamp parameters, such as, *start_time*, *end_time*, and *total_time* are scattered in the list. To view these parameters in order, the user can add the *normalize-time-labels* parameter to the *features.json* file.

To add the *normalize-time-labels* parameter in the *features.json* file, perform the following steps.

1. Login to the ESA CLI Manager.
2. Navigate to **Administration > OS Console**.
3. Navigate to the `/opt/protegility/alliance/config` directory.
4. Open the *features.json* file for editing.
5. Add the *normalize-time-labels* parameter in the *features.json* file. The *normalize-time-labels* parameter can be added in the following format:

```
{  
  "features": [  
    "normalize-time-labels"  
  ]  
}
```

The following table shows the default timestamp parameters and the normalized timestamp parameters.

Table 13-2: Default and Normalized timestamp parameters

Default Timestamp Parameters	Normalized Timestamp Parameters
auth_end_time	auth_time_end
auth_start_time	auth_time_start
auth_total_time	auth_time_total
end_time	time_end
start_time	time_start
total_time	time_total
pre_processing_time	time_pre_processing

The following snippet shows the transaction metrics with the Normalized timestamp parameters.

```
{
  "auth_cache_hit":false
  "auth_time_end":XXXXXXXXXX
  "auth_time_start":XXXXXXXXXX
  "auth_total_time":0.013525
  "auth_user_name":"admin"
  "client_ip":XXXXXXXXXX
  "dag_version":XXXXXXXXXX
  "fdt":37
  "http_outbound_ip": "192.168.1.100"
  "http_outbound_port": "443"
  "http_outbound_available_clients":100
  "http_outbound_count_new_connections":1
  "http_outbound_count_redirect":0
  "http_outbound_local_port":57494
  "http_outbound_max_queue_size":1200
  "http_outbound_size_download":15.00
  "http_outbound_size_queue":0
  "http_outbound_size_upload":15.00
  "http_outbound_speed_download":5305.00
  "http_outbound_size_header":10.000000
  "http_outbound_time_appconnect":0.000793
  "http_outbound_time_connect":0.000974
  "http_outbound_time_namelookup":0.000378
  "http_outbound_time_pretransfer":0.001021
  "http_outbound_time_starttransfer":0.001021
  "http_outbound_time_redirect":0.000000
  "http_outbound_time_request":0.003088
  "http_outbound_time_starttransfer":0.001025
  "http_outbound_time_total":0.002827
  "http_outbound_url": "http://sample.server.com:8081/echo/basicAuth"
  "http_reason_phrase": "OK"
  "http_status_code":200
  "learn_mode_enabled":false
  "local_port":49296
  "node_hostname":XXXXXXXXXX
  "node_id":1538
  "open_sessions":1
  "time_end":XXXXXXXXXX
  "time_pre_processing":0.000793
  "processing_time_downstream":0.013261
  "processing_time_request":0.035567
  "processing_time_start":XXXXXXXXXX
  "request_url": "http://sample-protected-server7.com:8081/echo/basicAuth"
  "service_name": "HTTP Service Tests"
  "service_type": "HTTP-GW"
  "time_end":XXXXXXXXXX
  "time_start":XXXXXXXXXX
  "time_total":0.026903
  "transformations": [
    {
      "replace_rules": [
        {
          "rule_name": "Replace Sensitive Data",
          "num_replace": 2
        }
      ]
    }
  ]
}
{
  "tunnel_name": "auto_8081"
  "user_name": "admin"
}
```

Figure 13-2: Transaction metrics with the normalized timestamp parameters

13.1 Total Time Breakdown for HTTP Request

This section describes the total time taken for processing the HTTP request.

The *total_time* value is calculated by adding the time taken by the following parameters:

- *time_pre_processing*: The time a HTTP or REST request waited before it was processed.
- *processing_time_request*: The time taken for the ruleset to process the request data.
- *processing_time_downstream*: The time taken to send a request to a downstream system and receive a response from the client.
- *processing_time_response*: The time taken for the ruleset to process the response data.

The following chart depicts the breakdown of the total time taken for a HTTP request.

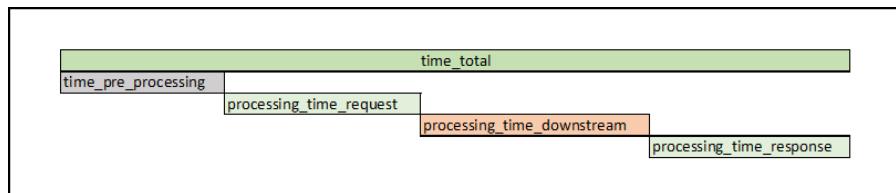


Figure 13-3: Total time breakdown for HTTP

The *processing_time_downstream* value is the difference between the start time of processing the response and the end time of processing a request. The *processing_time_downstream* is calculated by considering the time taken by any the following parameters:

- *http_outbound_time_queue*: The time that request spent in the queue before being processed.
- *http_outbound_namelookup*: The time taken to resolve name.
- *http_outbound_time_connect*: The time taken to connect to the remote host.
- *http_outbound_time_appconnect*: The time taken to complete the SSH/TLS handshake.

- *http_outbound_time_pretransfer*: The time from start until before the first byte is sent.
- *http_outbound_time_starttransfer*: The time taken from start of request until the first byte was received from server.
- *http_outbound_time_total*: Total time that the client library took to process the HTTP request.
- *http_outbound_time_redirect*: The time, in seconds, it took for all redirection steps including name lookup, connect, pretransfer, and transfer before the final transaction was started.

The following chart depicts the processing time downstream for a HTTP request.

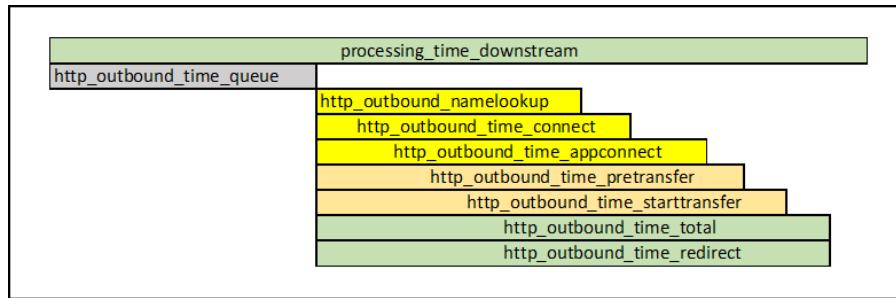


Figure 13-4: Processing time downstream breakdown for HTTP

Chapter 14

Error Metrics Logging

[14.1 Error Metrics with Non-Permissive Errors](#)

[14.2 Configuring the HTTP Status Codes](#)

Error Metrics allow a user to view details about errors, which are encountered while processing a file. The error metrics logging feature can be enabled at the service level.

When the **Protegility Data Protection** Transform Rule is configured, certain error conditions can occur during processing. The following are some of the error conditions:

- The input is too short or long for a particular data element
- Invalid Email ID
- Invalid Data Type
- Invalid Credit Card Details

When these error conditions occur, the rule will stop processing. The permissive error handling feature is hence used to handle the errors and process the erroneous input file.

For more information about permissive error handling, refer to the [Table: Protegility Data Protection Method](#).

If there are a lot of erroneous data in an input file, it can be difficult to identify and categorize errors. In that situation, the error metrics can be used to understand the total number of errors, the offset of where the error was encountered, the reasons why the error was encountered, the ruleset details, and so on.

Error metrics is written to the `gateway.log` file and the *Log Viewer* screen in the JSON format.

In the case of NFS and CIFS protocols, a lock file is created for each file that is to be processed. Error metrics will also be appended to the lock files alongside the update details for each file. For example, if there are ten files to be processed, namely, Test1 to Test10, then ten respective lock files will be created, namely, Test1.lock to Test10.lock. If there are any errors encountered in the Test1 file, then the error metrics for this file will be appended to the Test1.lock file.

Important:

The error metrics is only supported for the following payloads:

- CSV Payload
- Fixed Width

For more information about the **CSV** and **Fixed Width** payloads, refer to the sections [CSV Payload](#) and [Fixed Width](#).

Important:

The error metrics support is only available for the following services::

- HTTP
- REST
- NFS
- CIFS

Important:

The following conditions should be met to use the error metrics logging feature:

- Users must use the **Protegility Data Protection** method to transform the data.
- Permissive error handling should be configured at the transform rule.
- Error Metrics Logging field must be enabled at the service level.

Note: If the permissive error handling is disabled or a different transformation method is used, then the *total_error_count* will be 1 in the error metrics.

The sample error metrics for the REST request is as seen in the following log file snippet:

```
{
  "service_name": "Rest Error Reporting | Created | Verbose",
  "tunnel_name": "default_80",
  "start_time": "2019-05-04T04:59:13Z",
  "total_error_count": 7,
  "column_info": {
    "2": "first_name"
  },
  "reasons": [
    {
      "reason": "The input is too long (returnCode 0)",
      "rulesets": [
        {
          "ruleset": "first_name_ruleset",
          "offset": [
            {
              "columns": [
                {
                  "2": [
                    "rows": [
                      2,
                      5,
                      8,
                      11,
                      14,
                      17,
                      20
                    ],
                    "trimmed": false
                  ]
                }
              ],
              "error_count": 7
            }
          ]
        }
      ]
    }
  ],
  "request_uri": "http://restErrReporting1:8080//errorReporting201error",
  "id": "1b4112173f034b419ef5bffff9cb1de",
  "node_hostname": "protegility-cg312",
  "node_pid": 12779,
  "end_time": "2019-05-05T00:16Z",
  "total_time": "0:00:01.029595"
}
```

Figure 14-1: Sample Error Metrics

The following table describes the parameters available in the error metrics for different services.

Table 14-1: Error metrics parameters

Parameter	Supported on Services	Data type	Description
service_name	HTTP, REST, NFS, CIFS	string	The name of the service processing the request.
tunnel_name	HTTP, REST, NFS, CIFS	string	The name of the tunnel processing the request.



Parameter	Supported on Services	Data type	Description
request_uri	HTTP, REST	string	The URI of the request being processed by the DSG.
file_name	NFS, CIFS	string	The name of the file that is being processed by the DSG.
total_error_count	HTTP, REST, NFS, CIFS	integer	The total number of errors encountered while processing a request.
column_info	HTTP, REST, NFS, CIFS	integer and string	<p>For the CSV payload, a list of column numbers and column names will be logged when the errors are encountered.</p> <div style="background-color: #e0f2e0; padding: 10px;"> <p>Note:</p> <p>The Header will be taken from the CSV or the Fixed Width extract rule.</p> <p>While configuring the CSV extract rule, if the Header is set to <i>-1</i>, then the <i>column_info</i> will not be logged in the error metrics.</p> </div> <p>For the Fixed Width payload, a list of column numbers, column start, and column width will be logged when the errors are encountered.</p> <p>The following snippets shows the <i>column_info</i> parameter for the Fixed Width payload:</p> <pre>"column_info": { "1": { "column_start": 3, "column_width": 17 } },</pre>
reason	HTTP, REST, NFS, CIFS	string	The reason for a particular error will be displayed.
ruleset	HTTP, REST, NFS, CIFS	string	The traversal of the transform rule, which induces the error. For example, in the Figure-Sample Error Metrics the error is induced from the New Profile/HTTP Extract/Extract Rule/Rule rule.
columns	HTTP, REST, NFS, CIFS	integer	The column numbers where the error is encountered.
rows	HTTP, REST, NFS, CIFS	integer	The row numbers where the error is encountered.
trimmed	HTTP, REST, NFS, CIFS	boolean	True indicates that the error metrics is trimmed. The trimming of an error metrics depend on the <i>checkErrorLogAfterCount</i>

Parameter	Supported on Services	Data type	Description
			parameter, that is configurable in the <i>gateway.json</i> file. For more information about <i>checkErrorLogAfterCount</i> parameter, refer to the Table: gateway.json configurations .
error_count	HTTP, REST, NFS, CIFS	integer	The total number of errors encountered for a particular reason.
id	HTTP, REST, NFS, CIFS	integer	A unique ID to identify the transaction.
node_hostname	HTTP, REST, NFS, CIFS	string	The hostname of the DSG.
node_pid	HTTP, REST, NFS, CIFS	integer	The Process ID of the gateway process that processed request.
start_time	HTTP, REST, NFS, CIFS	string	The timestamp when the DSG received a request.
end_time	HTTP, REST, NFS, CIFS	string	The timestamp representing when a request was completed.
total_time	HTTP, REST, NFS, CIFS	string	The difference in seconds between the <i>end_time</i> and <i>start_time</i> parameters.

14.1 Error Metrics with Non-Permissive Errors

This section describes how the non-permissive errors are logged in the error metrics.

When the permissive error handling is disabled and an error is encountered while transforming data, it will capture the error once and stop processing the entire file. The error could be in the input file, ruleset, or any other configuration. In such scenarios, the error metrics will always be logged with a *total_error_count = 1*.

The sample error metrics with non-permissive errors are as seen in the following log file snippet:

```
{
  "service_name": "Rest Error Reporting | Created | Verbose",
  "tunnel_name": "default 80",
  "start_time": "2023-05-03T14:59:59.165168Z",
  "total_error_count": 1,
  "column_info": {
    "2": "first_name"
  },
  "reasons": [
    {
      "reason": "The input is too long (returnCode 0)",
      "rulesets": [
        {
          "ruleset": "_____",
          "offset": [
            {
              "columns": [
                {
                  "2": [
                    "rows": [
                      2
                    ],
                    "trimmed": false
                  ]
                }
              ],
              "error_count": 1
            }
          ]
        }
      ],
      "request_url": "http://restErrReporting1:8080//errorReporting201error",
      "id": "640daef89dd074cd0ab4eazc6df035ac",
      "node_hostname": "protegility-cg312",
      "node_pid": 12177,
      "end_time": "2023-05-04T00:39:07.42Z",
      "total_time": "0:00:01.225574"
    }
  ]
}
```

Figure 14-2: Sample Error Metrics with non-permissive errors

14.2 Configuring the HTTP Status Codes

This section describes how to configure the HTTP status codes for the errors that may occur while processing a file.

When errors are encountered and the user wants to handle them permissively, then different HTTP status codes can be configured in the *Error Code* field from the DSG Web UI. At the service level of the **RuleSet** page, an *Error Code* field is added for the **HTTP** and **REST** protocols to handle errors permissively.

Note: The **Error Code** field is not supported for the NFS and CIFS protocols.

The following are the HTTP status codes that can be configured from the Web UI:

- 200 OK
- 201 Created
- 202 Accepted
- 203 Non-Authoritative Information
- 205 Reset Content
- 206 Partial Content
- 400 Bad Request
- 401 Unauthorized
- 403 Forbidden
- 422 Unprocessable Entity
- 500 Internal Server Error
- 503 Service Unavailable

Note:

By default, the **Error Code** is set to *200 OK*.

The error metrics options for different protocols are as seen in the following figures:

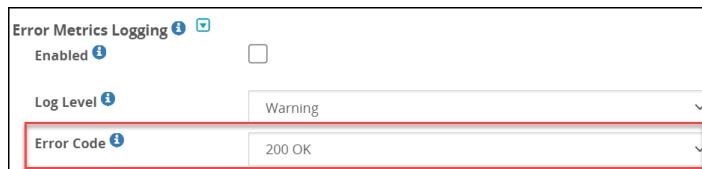


Figure 14-3: Error Metrics Logging options for HTTP and REST protocols



Figure 14-4: Error Metrics Logging options for NFS and CIFS protocols

Chapter 15

Best Practices

[15.1 Naming Convention](#)

[15.2 Learn Mode](#)

[15.3 Using Prefix and Suffix](#)

[15.4 Profile Reference](#)

[15.5 Modifying Out-of-the-box Profiles and Rules](#)

[15.6 Defining Services](#)

[15.7 Default Certificates and Keys](#)

[15.8 Migration of Data Tokenized by DSG](#)

This section provides information about best practices that can help you implement Data Security Gateway (DSG) better taking into considerations best approach to defining services, profile references, and more.

15.1 Naming Convention

It is recommended to keep the names of rules short but descriptive. Using the description field of a rules is highly recommended for the benefit of whomever might be maintaining it in the future. Rule names may appear in logs therefore keeping choosing an appropriate name would make it easier to find in the Ruleset tree.

Keeping the same host names, addresses and data element names across environment where possible will make it environment agnostic. Keeping the environments configuration in sync is common for testing and diagnosing. This can be done by backing up, export and restore the configuration of one environment to another. The minimum changes needed to be made after import and restore, the better.

15.2 Learn Mode

Learn mode is useful for studying the payload of an application for creating the appropriate data protection transformation rules, diagnosing a problem or analyzing performance. It however impacts performance due to the IO involved in the process. It also occupies disk space that otherwise wouldn't have been used. Some regulations may require certain handling of such log information in a production system which DSG may not comply with and therefore it is highly recommended to keep Learn mode disabled for the services used in a production system.

Note: The rules displayed in the *Learn mode* screen and the payload message difference are stored in a separate file in DSG. The default file size limit for this file is 3MB. If the payloads and rules in your configuration are high in volume, you can configure this file size.

In the ESA CLI Manager, navigate to **Administration > OS Console**. Edit the `webuiConfig.json` file in the `/opt/protegility/alliance/config/webinterface` directory to add the Learn mode file size parameter as follows:

```
{  
  "learnMode": {  
    "MAX_WORKER_THREADS": 5,  
    "LEARNSIZE_FLOW_DUMP_FILESIZE": 10
```

{}

15.3 Using Prefix and Suffix

The optional use of prefix and suffix to “mark” protected data makes the Ruleset more generic, optimized and resilient for potential modifications. This is since the unprotection set of rules is targeting a payload type rather than a specific message. This means that the rule will be useful for any new message which the application may be extended to so long that it uses the same payload type the system is configured to handle.

To make the use of prefix and suffix efficient, apply the data protection on a word by word bases rather than a complete sentence. The reason behind this recommendation is that application may use spaces to break down sentences to words and index each word separately. Using prefix and suffix on each word individually will maintain its value should the application do so in the backend.

By using different sequence for different class, prefix and suffix may also be used to distinguish between different types of protected data classification.

15.4 Profile Reference

Profile references can be used to point at a specific profile as if the rules in the profile existed in place of the reference. A single copy of repeatedly used rules hierarchy can then be maintained and referenced from other profiles.

For example, creating a generic rule to protect a SSN in a certain way might be useful in more than one place. It is recommended to create a separate profile and reference it from where it is needed. That way, should protection of SSN ever change in the future, adjustment will be required in one place only.

Note that references can be made across services. One may choose to create a service which is not associated with any tunnel (dormant) and use it as a container for profiles referenced from other services.

15.5 Modifying Out-of-the-box Profiles and Rules

DSG does not block users from updating any of the rules provided out-of-the-box. Customers are however encouraged to avoid such changes in profiles used in production environment. Updates that Protegility may release for these out-of-the-box profiles may conflict with customer modification. Customer modifications in this case may be overwritten. It is recommended to make a copy of the profile and disable the out-of-the-box profile/branch as a mean to avoid such conflict.

Note: Data is not lost even if profile is overwritten as a configuration backup copy is made with every patch/update install.

15.6 Defining Services

Services are fundamental containers that further define the rules used for data security operations. When designing DSG implementations, you must ensure that only one service type per protocol-hostname combination is created. You can further divide all applicable rules under this service in form of profiles.

If you create multiple services per protocol-hostname combination, DSG might never process the rules under the second service.

For example, consider a situation where you want certain payload, such as text, to use tokenization data elements as the chosen form of data security, and other payload, such as CSV, to use encryption data elements. You create Service A to use the tokenization elements, while Service B to use the encryption data elements. In the given scenario, if both services are enabled, DSG will execute the first service, Service A.



To avoid this situation, the correct approach would be creating a Profile A that includes subsequent rules to use tokenization data elements, and Profile B that includes rules to use encryption data elements.

Caution:

It is recommended that when creating Rulesets for Dynamic CoP, the *Profile Reference* rule is used for data transformation instead of the *Transform* rule. The security benefits of using *Profile Reference* rule are higher than the *Transform* rule since the requests can be triggered out of the secure network perimeter of an organization.

15.7 Default Certificates and Keys

The DSG generates default *admin* certificates and keys that are required for internal communication between DSG nodes and ESA.

If any certificates or keys in the DSG expire, then you must note the following points before regenerating them.

- It is recommended that you do not use any default certificates in production environments. Ensure that you use custom *admin* certificates and keys and upload them to the DSG using the ESA Web UI.
- For any other non-production environments, if you have used default *admin* certificates and keys, you must generate them from any machine other than the DSG.

The *admin* tunnel, which is used by DSG for internal communication between appliances (ESA and DSG), requires the **.pem** files. Ensure that for default DSG certificates, the **.pem** files are also generated along with the **.crt** and **.key** files.

Note:

Ensure that the *common name* is set to **protegilityClient** in the *openssl* command to generate the default certificates.

Note:

Ensure that after you generate the default certificates and keys, they are renamed as defined in the *.gateway.json* file. You can access the *gateway.json* file from the ESA Web UI, by navigating to **Settings > System**. The following image highlights the names of the certificates, keys, and commonName settings in the *gateway.json* file.

```
"admin": {
  "listenAddress": "ethMNG",
  "listenPort": 8585,
  "certificateFilename": "admin.pem",
  "certificateKeyFilename": "admin.key",
  "ciphers": "ECDH+AESGCM:DH+AESGCM:ECDH+AES256:DH+AES256:ECDH+AES128",
  "clientCACertificateFileName": "ca.pem",
  "clientCertificateFileName": "admin_client.pem",
  "clientCertificateKeyFileName": "admin_client.key",
  "commonName": "protegilityClient",
  "ssl_options": "{\"cert_reqs\":\"CERT_REQUIRED\"}"
},
```

Figure 15-1: *Gateway.json* key parameters

- If you have used custom certificates for the *admin* certificates and keys, then you must ensure that the *common name* that was used to generate the **.csr** file is set in the *gateway.json* file.

You must edit the `gateway.json` file to change the `commonName` parameter as required.

15.8 Migration of Data Tokenized by DSG

If you want to migrate any data tokenized in DSG, then there are some important considerations that must be noted.

Caution: The DSG only generates UTF-8 encoded tokens. If the tokens generated by DSG need to be used in other ecosystems that are configured for an encoding other than UTF-8, a translation is required to the encoding of the target ecosystem.

Chapter 16

RuleSet Reference

[*16.1 Services*](#)

[*16.2 Profiles*](#)

[*16.3 Rules*](#)

The top level object defined under Ruleset is called a Service. A Service represents one or more applications by housing profiles. Profiles are container for rules, and rules are applied on messages transmitted through Data Security Gateway (DSG).

16.1 Services

DSG supports multiple protocols that are defined as services in the Ruleset hierarchy.

In the DSG, the following service types are available:

- **REST API Service:** The DSG acts as a REST API Server, protecting or unprotecting application in a trusted domain.
- **Gateway Service:** The DSG acts as a gateway to protect sensitive information before it reaches an untrusted domain.

You can choose the following service based on the protocols configured. The different Gateway Service types are as follows:

- REST API
- HTTP
- WebSocket Secure (WSS)
- SMTP
- SFTP
- Amazon s3
- Mounted File System

16.1.1 Gateway Service Fields

Some fields are common across the multiple protocol types that are defined as services.

The following figure illustrates all the common fields for the available service types.

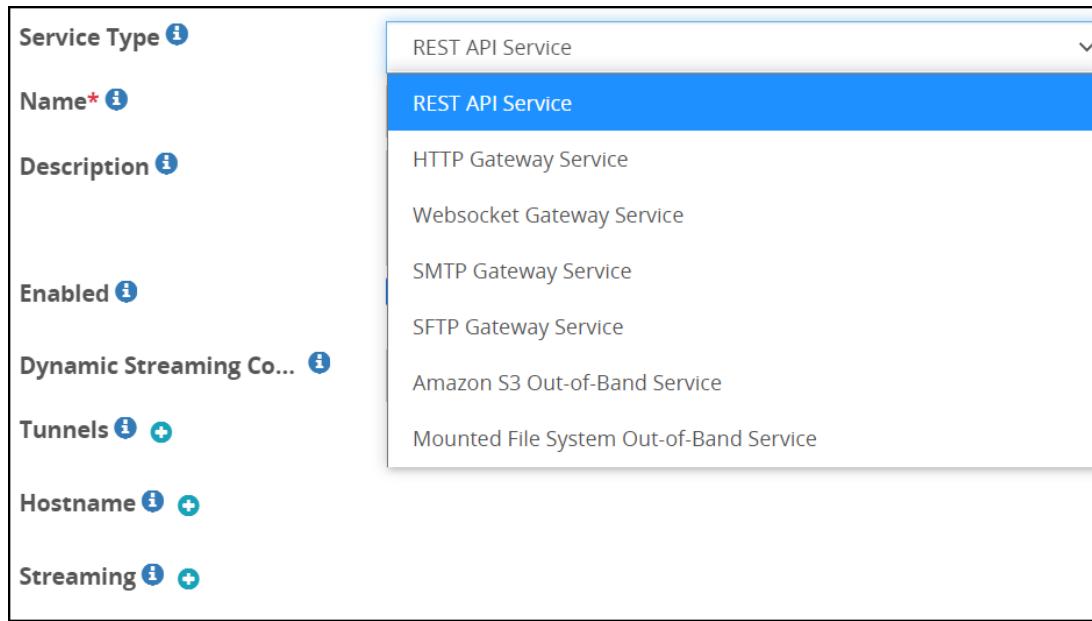


Figure 16-1: Gateway Service Fields

The following table describes all the common fields for the available Service Types.

Table 16-1: Service Type Fields

Field	Description	
Service Type	Specify the role of this service i.e. whether to act as REST API or act as a gateway for a specific protocol.	
Name	Name for the Service.	
Description	Description for the Service.	
Enabled	Enable or disable the Service.	
Tunnels	List of tunnels lying below the service instance.	
Hostnames	List of hostname to forwarding address mappings	
	Hostname	Hostname or the IP address for an inbound request received by the gateway.
	Forwarding Address	Hostname or the IP address for an outbound request forwarded by the gateway.
Password Masking	List of parameters value to be masked before the output is sent to the log files.	
	Pattern	Regular expression to find text to replace in the parameter.
	Resource	Regular expression to look for in the parameter before masking it.
	Mask	The replacement text which acts as a mask for the pattern.
Learn Mode Settings	Filters for capturing details to be presented in the learn mode.	
	Enabled	Enable or disable learn mode settings.
	Exclude Resource	Values in the field are excluded from the Learn Mode logging.

Field	Description	
	Exclude Content Type	Content type specified in the field is excluded from the Learn Mode logging.
	Include Resource	Values in the field are included in the Learn Mode logging.
	Include Content-Type	Content type specified in the field is included in the Learn Mode logging.
Transaction Metrics Logging		Define if you want to log detailed transaction metrics, such as, protect operation performed, length of the data, service used to perform protection, tunnel used, and so on.
	Enabled	Enable or disable transaction metrics to be logged in the log file.
	Log Level	Select from the following logging levels: <ul style="list-style-type: none"> • Warning • Information • Verbose <p>Note: Ensure that the log level you select is the same or part of a higher log subset that you defined in the gateway log level.</p>
Transaction Metrics in HTTP Response Header*		
	HTTP Response Header Reporting Enabled	Enable or disable detailed transaction metrics such as, data security operation performed, length of the data, service used to perform protection, tunnel used, and so on in the HTTP Response Header. <p>Note: If the HTTP Response Header Reporting Enabled option is selected and streaming is enabled, the transaction metrics data will not be displayed in the HTTP Response Header.</p>
	HTTP Response Header Name	Name of the HTTP Response Header carrying the transaction metrics data. The default value for this option is X-Protegility-Transaction-Metrics. You can change the default value as per your requirements. <p>Note: The name of the HTTP Response Header must be defined with valid characters. An HTTP Response Header name defined with invalid characters is automatically modified to the default value X-Protegility-Transaction-Metrics.</p>

* -The Transaction Metrics in HTTP Response Header option is only available for the REST API and HTTP services.

16.1.2 REST API

The fields related to REST API service are described in this section.

The fields for the REST API service are as seen in the following figure.



Service Type <i>i</i>	REST API Service
Name* <i>i</i>	
Description <i>i</i>	
Enabled <i>i</i>	<input checked="" type="checkbox"/>
Dynamic Learn Mode H... <i>i</i>	
Dynamic Streaming Co... <i>i</i>	
Tunnels <i>i</i> <i>+</i>	
Hostname <i>i</i> <i>+</i>	
Streaming <i>i</i> <i>+</i>	
Password Masking <i>i</i> <i>+</i>	
Learn Mode Settings <i>i</i>	
Enabled <i>i</i>	<input type="checkbox"/>
Exclude Resource <i>i</i>	\.(css png gif jpg ico woff ttf svg eot)(\? \b)
Exclude Content-Type <i>i</i>	\bcss image video svg\b
Include Resource <i>i</i>	
Include Content-Type <i>i</i>	
Credits <i>i</i>	250000000
Asynchronous Client Configuration <i>i</i>	
Authentication Cache ... <i>i</i>	900
Performance Measurements <i>i</i>	
Error Metrics Logging <i>i</i>	
Enabled <i>i</i>	<input type="checkbox"/>
Log Level <i>i</i>	Warning
Error Code <i>i</i>	200 OK
Transaction Metrics Logging <i>i</i>	
Enabled <i>i</i>	<input type="checkbox"/>
Log Level <i>i</i>	Warning
Transaction Metrics in HTTP Response Header <i>i</i>	
<input type="button" value="Cancel"/> <input type="button" value="Save"/>	

Figure 16-2: REST API Gateway Specific Fields

The following table describes the additional fields for the REST API Gateway service.

Table 16-2: REST API Gateway Specific Fields

Field	Sub-Field	Description	Default (if any)
Dynamic Learn Mode Header		The header that will be used to send a request to enable the learn mode for a particular URI.	
Dynamic Streaming Configuration*		HTTP header that will be used to send a request.	
Streaming		<p>Enabling streaming lets you process a payload in smaller chunks that are broken based on delimiters defined and processed as they are chunked. Using streaming, you no longer must wait for the entire payload to process, and then transmitted. The chunk size must be entered in bytes.</p> <p>List of file processing delimiters to process file using streaming.</p> <div style="background-color: #e0f2e0; padding: 10px;"> Note: The Text, CSV, and Binary payloads are supported. If you want to use XML/JSON payload with streaming, ensure you use the Text payload for extract rule. </div>	Chunk size - 65536
Authentication Cache Timeout		Define the amount of time for which the username and password in the REST request is stored in cache.	900 seconds
Asynchronous Client Configuration		If streaming is enabled and you plan to use an asynchronous HTTP client, then these settings must be configured. The DSG is optimized to handle asynchronous requests.	
		<div style="background-color: #e0f2e0; padding: 10px;"> Note: This parameter is applicable only with REST streaming. </div>	
	HTTP Async Client Enabled	<p>Select to enable when HTTP asynchronous client will send a request to DSG.</p> <div style="background-color: #e0f2e0; padding: 10px;"> Note: The <i>HTTP Async Client Header Name</i> header must be sent as part of the HTTP request for DSG to understand that the incoming requests are sent from an asynchronous client. If the header is not sent as part of the request, then the DSG assumes that the request is sent from a synchronous client. </div> <div style="background-color: #e0f2e0; padding: 10px;"> Note: This parameter is applicable only with REST streaming. </div>	False
	HTTP Async Client Header Name	<p>Provide the header name that must be set in an HTTP request in the client such that DSG understands that the request is sent from an asynchronous HTTP client.</p> <p>For example, if the header name is set to <i>X-Protegility-Async-Client</i> in the service, then when a request is sent to the DSG, the header value must be set to either 'yes', 'true', or '1'.</p> <div style="background-color: #e0f2e0; padding: 10px;"> Note: </div>	n/a



Field	Sub-Field	Description	Default (if any)
		This parameter is applicable only with REST streaming.	
Error Metrics Logging		Enable if you want to log the detailed error metrics, such as, the total number of errors, the offset of where the error was encountered, the reasons why the error was encountered, the ruleset details, and so on.	
	Enabled	Enable or disable error metrics to be logged in the log file.	
	Log Level	<p>Select from the following logging levels:</p> <ul style="list-style-type: none"> • Warning • Information • Verbose <p>Note: Ensure that the log level you select is the same or part of a higher log subset that you defined in the gateway log level.</p>	
	Error Code	<p>Set one HTTP status code for the errors that may occur in the file while processing it.</p> <p>Select from the following HTTP status codes:</p> <ul style="list-style-type: none"> • 200 OK • 201 Created • 202 Accepted • 203 Non-Authoritative Information • 205 Reset Content • 206 Partial Content • 400 Bad Request • 401 Unauthorized • 403 Forbidden • 422 Unprocessable Entity • 500 Internal Server Error • 503 Service Unavailable <p>For more information about Error Code field, refer to the section Configuring the HTTP Status Codes.</p>	200 OK

* -The dynamic streaming configuration can be explained as follows:

If you want to send dynamic requests to enable streaming on a given URI, you can use this field. Consider an example, where you set this value as *X-Protegility-Rest-Header*. When you send an HTTP request with the X-Protegility-Rest-Header header value, DSG will begin the data protection for that URI based on the parameters provided in the request.

A typical format for the value in the header is as follows:

```
"{ "streaming": { "uri": "/echo", "delimiter": "(?ms)(^.*\r?\n)", "chunk_size": 5000 } }"
```

Table 16-3:

Parameter	Description	Default
delimiter	Regular Expression used to delimit stream. Rules will be invoked on delimited streams.	(?ms)(^.*\r?\n)

Note:



Parameter	Description	Default
	If the delimiter value is not matched, then the data will be processed in non-streaming mode.	
Uri	Regular Expression to look for in the payload before applying streaming (e.g. \.csv\$). Streaming is applied only to requests where URI matches the regex pattern.	
chunk_size	Size of the smaller chunks that the data must be broken into. The chunk size must be entered in bytes.	65536

Note: The delimiter parameter must be sent as part of the HTTP header information. The uri and chunk_size parameters are optional. If uri is not provided, the request URI is considered, while if the chunk_size is not provided, the chunk size defined in HTTP tunnel configuration is considered.

16.1.3 HTTP Gateway

The fields related to HTTP service are described in this section.

The fields for the HTTP Gateway service are as seen in the following figure.

Service Type <i>i</i>	HTTP Gateway Service
Name* <i>i</i>	HTTP
Description <i>i</i>	
Enabled <i>i</i>	<input checked="" type="checkbox"/>
Dynamic Learn Mode H... <i>i</i>	
Tunnels <i>i</i> <i>+</i>	
Host Addresses <i>i</i> <i>+</i>	
Auto Handle Domain N... <i>i</i>	<input checked="" type="checkbox"/>
Outbound Transport S... <i>i</i>	
Password Masking <i>i</i> <i>+</i>	
Learn Mode Settings <i>i</i>	
Enabled <i>i</i>	<input type="checkbox"/>
Exclude Resource <i>i</i>	\.(css png gif jpg ico woff ttf svg eot)(\? \b)
Exclude Content-Type <i>i</i>	\bccss image video svg\b
Include Resource <i>i</i>	
Include Content-Type <i>i</i>	
Credits <i>i</i>	250000000
Authentication Cache ... <i>i</i>	900
Performance Measurements <i>i</i> <i>▼</i>	
Enabled <i>i</i>	<input type="checkbox"/>
Error Metrics Logging <i>i</i> <i>▼</i>	
Enabled <i>i</i>	<input type="checkbox"/>
Log Level <i>i</i>	Warning
Error Code <i>i</i>	200 OK
Transaction Metrics Logging <i>i</i> <i>▼</i>	
Enabled <i>i</i>	<input type="checkbox"/>
Log Level <i>i</i>	Warning
Transaction Metrics in HTTP Response Header <i>i</i> <i>▼</i>	
<input type="button" value="Cancel"/> <input type="button" value="Save"/>	

Figure 16-3: HTTP Gateway-specific fields

The following table describes the additional fields for the HTTP Gateway service.

Table 16-4: HTTP Gateway Specific Fields

Field	Sub-Field	Description	Default (if any)
Dynamic Learn Mode Header		The header that will be used to send a request to enable the learn mode for a particular URI.	
Auto Handle Domain Name Rewrite		Adds the domain name, rewrites the filters and the rules that replace the host name in the forwarded requests or responses as per the target or source hostname.	
Outbound Transport Settings		Name-Value pairs used with the outbound transport.	
Authentication Cache Timeout		Define the amount of time for which the username and password in the REST request is stored in cache.	900 seconds
Error Metrics Logging		Enable if you want to log the detailed error metrics, such as, the total number of errors, the offset of where the error was encountered, the reasons why the error was encountered, the ruleset details, and so on.	
	Enabled	Enable or disable error metrics to be logged in the log file.	
	Log Level	<p>Select from the following logging levels:</p> <ul style="list-style-type: none"> • Warning • Information • Verbose <div style="background-color: #e0f2e0; padding: 5px; margin-top: 10px;"> Note: Ensure that the log level you select is the same or part of a higher log subset that you defined in the gateway log level. </div>	
	Error Code	<p>Set one HTTP status code for the errors that may occur in the file while processing it.</p> <p>Select from the following HTTP status codes:</p> <ul style="list-style-type: none"> • 200 OK • 201 Created • 202 Accepted • 203 Non-Authoritative Information • 205 Reset Content • 206 Partial Content • 400 Bad Request • 401 Unauthorized 	200 OK



Field	Sub-Field	Description	Default (if any)
		<ul style="list-style-type: none"> • 403 Forbidden • 422 Unprocessable Entity • 500 Internal Server Error • 503 Service Unavailable <p>For more information about Error Code field, refer to the section Configuring the HTTP Status Codes.</p>	

The options for the Outbound Transport Settings field in the HTTP Gateway are described in the following table.

Options	Description	Default
auth_mode	Authentication mode “digest” or “basic” (default)	basic
auth_username	Username for HTTP authentication	n/a
auth_password	Password for HTTP authentication	n/a
connect_timeout	Timeout for initial connection (in seconds)	300
request_timeout	Timeout for the entire request (in seconds)	180
network_interface	Network interface used for the request	n/a
proxy_host*	HTTP Proxy hostname	n/a
proxy_port*	HTTP Proxy port	n/a
proxy_username*	HTTP Proxy username	n/a
proxy_password*	HTTP Proxy password	n/a
validate_cert	Validation required for the server certificate (false).	n/a
ca_certs	Filename of the CA certificates (in PEM format only)	n/a
client_key	Filename for the client SSL key	n/a
client_cert	Filename for the client SSL certificate	n/a
allow_nonstandard_methods	Flag to allow nonstandard HTTP method constructions such as HTTP PUT method with no body content.	false
max_599_retries	Maximum number of retries for the gateway to send a request to the server when an status code of 599 occurs.	<p>1 minimum: 1 and maximum: 5</p> <p>Note: If you set the value of the "max_599_retries" less than 1, the value is automatically set to 1. If you set the value of the "max_599_retries"</p>

Options	Description	Default	
		greater than 5, the value is automatically set to 5.	
Error Metrics Logging	Enable if you want to log the detailed error metrics, such as, the total number of errors, the offset of where the error was encountered, the reasons why the error was encountered, the ruleset details, and so on.		
	Enabled	Enable or disable error metrics to be logged in the log file.	
	Log Level	<p>Select from the following logging levels:</p> <ul style="list-style-type: none"> • Warning • Information • Verbose <p>Note: Ensure that the log level you select is the same or part of a higher log subset that you defined in the gateway log level.</p>	
	Error Code	<p>Set one HTTP status code for the errors that may occur in the file while processing it.</p> <p>Select from the following HTTP status codes:</p> <ul style="list-style-type: none"> • 200 OK • 201 Created • 202 Accepted • 203 Non-Authoritative Information • 205 Reset Content • 206 Partial Content • 400 Bad Request • 401 Unauthorized • 403 Forbidden • 422 Unprocessable Entity • 500 Internal Server Error • 503 Service Unavailable <p>For more information about Error Code field, refer to the section Configuring the HTTP Status Codes.</p>	200 OK

* - The DSG uses the proxy parameters to forward the outgoing requests from DSG through the proxy. The DSG supports any proxy that can handle *HTTP CONNECT* method for tunneling HTTPS connections.



The following snippet describes the format for the Outbound Transport Settings field:

```
{
  "connect_timeout": 100,
  "request_timeout": 100,
  "ca_certs": "/opt/cert.cer"
  "max_599_retries":4
}
```

In the above example, the connect timeout is set as 100 seconds, the request timeout is set as 100 seconds and the path for the CA certificate is specified.

The curl is a command-line tool to transfer data between servers. This tool supports protocols, such as HTTP, HTTPS, FTP, FTPS, GOPHER, DICT, LDAP, TELNET, FILE, and so on. The features supported by curl include user authentication, proxy support, cookies, HTTP post, FTP upload, file transfer resume, SSL connections, and so on. The Gateway uses curl stack for outbound HTTP transport. You can use the curl options in the **Outbound Transport Settings** field.

Options	Description	Default
RESOLVE	Allows to resolve a request hostname to an IP address.	n/a
TCP_NODELAY	Setting this option disables the TCP Nagle algorithm. The field accepts the following two values: <ul style="list-style-type: none"> • 1 = Set • 0 = Clear The Nagle algorithm minimizes the number of small packets on the network.	1

The format for the curl options in the **Outbound Transport Settings** field is shown in the following example. In this example, the hostname `www.protegility.com` resolves to the IP address `2.10.1.4` while the `hostname else.domain.com` resolves to the IP address `2.10.2.3`. The `TCP_NODELAY` option is set to 1 indicating that the Nagle's algorithm is disabled.

```
{
  "network_interface": "ethMNG",
  "validate_cert": false,
  "options": {
    "RESOLVE": [
      "www.protegility.com:443:2.10.1.4",
      "else.domain.com:443:2.10.2.3"
    ],
    "TCP_NODELAY": 1
  }
}
```

16.1.4 Secure WebSocket (WSS) Service

The Secure WebSocket (WSS) service is a communication protocol that provides bi-directional communication between a client and a server over a single established connection.

In the DSG, the WSS service can be used by configuring the HTTP Tunnel. The WSS service is designed for listening to traffic on HTTP and HTTPS ports `80` and `443` respectively.

Caution:

In this release, the DSG uses the WSS service to pass through data as-is without performing any data protection operation, such as, protect, unprotect, and reprotect. You cannot invoke any child rules using the WSS service.

The fields for the WSS Gateway service are as seen in the following figure.

The screenshot shows the configuration interface for a WSS Gateway Service. The top section is titled "Service Type" and shows "Websocket Gateway Service" selected. Below it is a "Name" field, which is mandatory (*). The "Description" field is empty. The "Enabled" checkbox is checked. Under "Tunnels", there is a plus sign icon. The "Host Addresses" and "URLs" sections each have a plus sign icon. The "Origin Checking" dropdown is set to "SameDomain". The "Auto Handle Domain N..." checkbox is checked. The "Outbound Transport S..." section is empty. The "Password Masking" section has a plus sign icon. The "Learn Mode Settings" section has a dropdown arrow. The "Authentication Cache ... 900" section contains the value "900". The "Performance Measurements" and "Transaction Metrics Logging" sections both have dropdown arrows. The final section, "Transaction Metrics in HTTP Response Header", also has a dropdown arrow.

Figure 16-4: WSS Gateway Specific Fields

The following table describes the additional fields for the WSS Gateway service.

Table 16-5: WSS Gateway Specific Fields

Field	Sub-Field	Description	Default (if any)
URI		List the required URI to receive the request.	
Origin Checking		Checks the websocket handshake origin header.	
Auto Handle Domain Name Rewrite		Adds the domain name, rewrites the filters and the rules that replace the host name in the forwarded requests or responses as per the target or source hostname.	
Outbound Transport Settings		Name-Value pairs used with the outbound transport.	

Field	Sub-Field	Description	Default (if any)
Authentication Cache Timeout		Define the amount of time for which the username and password in the REST request is stored in cache.	900 seconds

16.1.5 SMTP Gateway

The SMTP Gateway service provides options that must be configured to define the level of extraction that must be performed on the incoming requests on the DSG. Based on the requirements, data security operations are performed on the extracted sensitive data.

The fields for the SMTP Gateway service are as shown in the following figure.

The screenshot shows a configuration interface for an 'SMTP Gateway Service'. It includes fields for Name, Description, Enabled status, Tunnels, Hostnames, and Outbound Transport Settings. The 'Outbound Transport Settings' field is specifically highlighted with a red box, indicating it is a key configuration area for this service type.

Figure 16-5: SMTP Gateway-specific fields

The following table describes the additional fields for the SMTP Gateway service.

Table 16-6: SMTP Gateway Specific Fields

Field	Sub-field	Description
Hostnames	Host Address	Hostname or the IP address for an inbound request received by the gateway. The service IP of the DSG must be specified. For example, secured-smtp.abc.com .
	Forwarding Address	Hostname or the IP address for an outbound request forwarded by the gateway. The hostname or IP address of the SMTP server must be specified. For example, smtp.abc.com .
Outbound Transport Settings		Name-Value pairs used with the outbound transport.

The *ssl_options* supported for the Outbound Transport Settings in the SMTP Gateway are described in the following table.

Table 16-7: Outbound Transport Settings

Options	Description	Default
certfile	Path of the certificate stored in the DSG to be sent to the SMTP server.	n/a
keyfile	Path of the key stored in the DSG to be sent to the SMTP server.	n/a
cert_reqs	Specifies whether a certificate is required for validating the TLS/SSL connection between the DSG and the SMTP server. The following values can be configured: <ul style="list-style-type: none"> • CERT_NONE: If the parameter is set to <i>CERT_NONE</i>, then the SMTP server certificate is not required for validating the SSL connection between the DSG and the SMTP server. • CERT_OPTIONAL: If the parameter is set to <i>CERT_OPTIONAL</i>, then the SMTP server certificate is not required for validating the SSL connection between the DSG and the SMTP server. The SMTP server certificate is validated only if it is provided. • CERT_REQUIRED: If the parameter is set to <i>CERT_REQUIRED</i>, then the SMTP server certificate is required for validating the SSL connection between the DSG and the SMTP server. 	<i>CERT_NONE</i>
ssl_version	Specifies the SSL protocol version used for establishing the SSL connection between the DSG and the SMTP server.	<i>PROTOCOL_SSLv23</i>
ciphers	Specifies the list of supported ciphers.	<i>'ECDH+AESGCM','DH+AESGCM','ECDH+AES256','DH+AES256','ECDH+AES128','DH+AES','RSA+AESGCM','RSA+AES'</i>
ca_certs	Path where the CA certificates (in PEM format only) are stored.	n/a

16.1.6 SFTP Gateway

The SFTP Gateway service can be implemented with either Password authentication or Public Key exchange authentication.

The fields for the SFTP Gateway service are as seen in the following figure.

The screenshot shows the configuration interface for an SFTP Gateway service. The 'Service Type' is set to 'SFTP Gateway Service'. The 'Name' is 'SFTP' and 'Enabled' is checked. Under 'Tunnels', there is one entry: 'SFTP_2222'. In the 'Host Addresses' section, there is one host address with a forwarding address of '10.10.3.6'. The 'User Authentication Method' is set to 'Password'. Under 'Learn Mode Settings', 'Enabled' is checked, and the excluded resources and content types are listed as follows:

Exclude Resource	Value
Exclude Resource	\(css png gif jpg ico woff ttf svg eot)(\? \b)
Exclude Content-Type	\bcss\b Image\b video\b svg\b

At the bottom right are 'Cancel' and 'Save' buttons.

Figure 16-6: SFTP Gateway-specific fields

The additional fields for the SFTP Gateway service when authentication method is Public Key are as seen in the following figure.

Service Type i
Name* i
Description i
Enabled i
Tunnels i +
Host Addresses i +
Host Address* i
Forwarding Address* i
Streaming i +
Filename* i
Delimiter* i
User Authentication Method i
Inbound Push Public Keys file* i
Outbound Push Private Key file* i
Outbound Push Private Keys file pas... i
Outbound Transport Settings i

Cancel Save

Figure 16-7: SFTP-specific fields- Public Key Authentication

Before you begin

Ensure that the following pre-requisites are complete before you start using the SFTP gateway with Public Key authentication method.

Caution: The DSG supports RSA keys. Ensure that only RSA keys are uploaded to the ESA/DSG Web UI.

- The SFTP client *Public Key* must be available and upload to the **Certificates** screen in the ESA Web UI.
 - The DSG *Public Key* and *Private Key* must be generated and uploaded to the **Certificates** screen in the ESA Web UI.
 - The DSG *Public Key* must be uploaded to the SFTP server.
- Ensure that the DSG Public Key is granted [644](#) permissions on the SFTP server.

The following table describes the additional fields relevant for the SFTP Gateway service.

Note: SFTP tunnel automatically sets the user identity with an authenticated username. Thus, subsequent calls to Protegility Data Protection transformations actions are done on behalf of the authenticated user.

Important:

If you are using SFTP with DSG, then the following SFTP commands are not supported.

- [df](#)
- [chgrp](#)
- [chown](#)

Table 16-8: SFTP Gateway Specific Fields

Field	Sub-Field	Description	Default (if any)
Streaming		<p>Enabling streaming lets you process a payload in smaller chunks that are broken based on delimiters defined and processed as they are chunked. Using streaming, you no longer must wait for the entire payload to process, and then transmitted.</p> <p>List of file processing delimiters to process file using streaming.</p> <p>Note: The Text, CSV, and Binary payloads are supported. If you want to use XML/JSON payload with streaming, ensure you use the Text payload for extract rule.</p>	Chunk size - 32 kB If you want to change the chunk size, modify the <code>default_max_packet_size</code> parameter in the Inbound Settings for the tunnel.
	Filename	<p>Regular Expression to look for in the payload before applying streaming (e.g. <code>\.csv\$</code>). Streaming is applied only to requests where URI matches the regex pattern.</p> <p>Note: Click Test Regex to verify if the regex expression is valid.</p>	
	Delimiter	<p>Regular Expression used to delimit stream. Rules will be invoked on delimited streams.</p> <p>Note: Click Test Regex to verify if the regex expression is valid.</p> <p>Note: If the delimiter value is not matched, then the data will be processed in non-streaming mode.</p>	
User Authentication Method		SFTP authentication method used to communicate between client and server.	
	Password	Enables password authentication for communication. You must enter password, when prompted, while initiating connection with the SFTP server.	
	Public Key	<p>Enable Public Key method for communication. The SFTP client shares its Public Key with the gateway and the gateway shares its Public Key with the SFTP server. This enables password-less communication between SFTP client and server when gateway is the intermediary.</p> <p>Ensure that the pre-requisites are completed before you start using the SFTP gateway.</p>	
	Inbound Push Public Keys file	<p>Specifies the file name for the SFTP client Public Key.</p> <p>Note: For more information on SFTP Gateway specific fields, refer to the section SFTP Gateway with Public Key Authentication.</p>	



Field	Sub-Field	Description	Default (if any)
	Outbound Push Private Key file	<p>Specifies the file name for the Gateway Private Key.</p> <p>Note: For more information on SFTP Gateway specific fields, refer to the section SFTP Gateway with Public Key Authentication.</p>	
	Outbound Push Private Keys file passphrase	Enter the passphrase for DSG Private Key. If no value is entered for encrypting the private key, the passphrase value is null .	
Outbound Transport Settings		Additional outbound settings that you want to parse during SFTP communication.	

The options for the Outbound Transport Settings field in the SFTP Gateway are described in the following table.

Table 16-9: Outbound Transport Settings

Options	Description	Default (if any)
window_size	SSH Transport window size. The datatype for this option is bytes.	3145728
use_compression	Toggle SSH transport compression.	TRUE
max_request_size	Set the maximum size of the message that is sent during transmission of a file. The maximum limit for servers that accept message size more than the default value is 250 KB.	32768
enable_setstat	Set to <i>False</i> when using the AWS Transfer for SFTP as the SFTP server.	True

16.1.7 Amazon S3 Gateway

The fields related to Amazon S3 service are described in this section.

The additional fields for the Amazon S3 Gateway service are as seen in the following figure.

The screenshot shows the configuration interface for an Amazon S3 Out-of-Band Service. It includes sections for basic service settings, object mapping, streaming rules, and transport configurations.

- Service Type:** Amazon S3 Out-of-Band Service
- Name:** S3 service
- Description:** *
- Enabled:** Checked
- Tunnels:** s3 Tunnel
- Object Mapping:**
 - Source:** john.doe/incoming
 - Target:** john.doe/outgoing
- Streaming:**
 - Object Key:** \.*?
 - Delimiter:** (?ms)(^.*\r?\n?)
- Outbound Transport Settings:** {"sse-customer-algorithm": "AES256"}
- Learn Mode Settings:**
- Enabled:** Unchecked

Figure 16-8: Amazon S3-specific fields

The following table describes the additional fields relevant for the Amazon S3 Gateway service.

Table 16-10: Amazon S3 Gateway Specific Fields

Field	Sub-Field	Description
Object Mapping		List of source and target objects that the service will use.
	Source	Bucket path where data that needs to be protected is stored. For example, <i>john.doe/incoming</i> . Note: The DSG supports four levels of nested folders in an Amazon S3 bucket.
	Target	Bucket path where protected data is stored. For example, <i>john.doe/outgoing</i> .
Streaming		List of file processing delimiters to process file using streaming. Note: The Text, CSV, and Binary payloads are supported. If you want to use XML/JSON payload with HTTP streaming, ensure you use the Text payload for extract rule.
	Filename	Regular Expression to look for in the file's name and path before applying streaming (e.g. \.csv\$)
	Delimiter	Regular Expression used to delimit stream. Rules will be invoked on delimited streams. Note:

Field	Sub-Field	Description
		If the delimiter value is not matched, then the data will be processed in non-streaming mode.

The options for the **Outbound Transport Settings** field in the Amazon S3 Gateway are described in the following table.

Table 16-11: Amazon S3 Outbound Transport Settings

Options	Description
SSECustomerAlgorithm	If server-side encryption with a customer-provided encryption key was requested, the response will include this header confirming the encryption algorithm used.
SSECustomerKey	Constructs a new customer provided server-side encryption key.
SSECustomerKeyMD5	If server-side encryption with a customer-provided encryption key was requested, the response will include this header to provide round trip message integrity verification of the customer-provided encryption key.
ServerSideEncryption	The Server-side encryption algorithm used when storing this object in S3 (e.g., AES256, aws:kms).
StorageClass	Specifies constants that define Amazon S3 storage classes.
SSEKMSKeyId	Specifies the ID of the AWS Key Management Service (KMS) master encryption key that was used for the object.
ACL	Allows controlling the ownership of uploaded objects in an S3 bucket. For example, if ACL or Access Control List is set to " <i>bucket-owner-full-control</i> ", new objects uploaded by other AWS accounts are owned by the bucket owner. By default, the objects uploaded by other AWS accounts are owned by them.

16.1.8 Mount File System Out-of-Band Service

The fields related to mounted file system services, namely NFS and CIFS are described in this section.

The additional fields for the Mounted File System service are as seen in the following figure.

Service Type i Mounted File System Out-of-Band Service

Name* i NFS/CIFS Service

Description i

Enabled i

Tunnels i + NFS_BASIC

File Mapping i +

Streaming i +

Outbound Transport S... i

Learn Mode Settings i

Enabled i

Exclude Resource i \.(css|png|gif|jpg|ico|woff|ttf|svg|eot)(\?|\b)

Exclude Content-Type i \bcss|image|video|svg\b

Include Resource i

Include Content-Type i

Credits i 250000000

Error Metrics Logging i ▼

Enabled i

Log Level i Warning

Transaction Metrics Logging i ▼

Enabled i

Log Level i Warning

Cancel Save

Figure 16-9: Mounted File System Specific Fields

The following table describes the additional fields relevant for the Mounted File System service.

Table 16-12: Mounted File System Specific Fields

Field	Sub-Field	Description
File Mapping	Source	List of source and target files that the service will process. Regex logic that includes the source path where data that needs to be protected is stored along with the filter to identify specific files.

Field	Sub-Field	Description
		<p>For example, if you set <code>(.*\//)input\//(.*)</code> as the value, all the files in the input folder will be selected for processing.</p> <p>Note: Click Test Regex to verify if the regex expression is valid.</p>
	Target	<p>Regex logic that includes the target path where processed data is stored along with other identifiers, such as appending additional tag.</p> <p>For example, if you set <code>\output\2.processed</code> as the value, the processed files will move to the <code>I/output</code> folder with <code>.processed</code> appended to them.</p> <p>Note: Click Test Regex to verify if the regex expression is valid.</p>
Streaming		<p>Enabling streaming lets you process a payload in smaller chunks that are broken based on delimiters defined and processed as they are chunked. Using streaming, you no longer must wait for the entire payload to process, and then transmitted.</p> <p>List of file processing delimiters to process file using streaming.</p> <p>Note: The Text, CSV, and Binary payloads are supported. If you want to use XML/JSON payload with streaming, ensure you use the Text payload for extract rule.</p>
	File Key	<p>Regular Expression to look for in the payload before applying streaming (e.g. <code>\.csv\$</code>). Streaming is applied only to requests where File Key matches the regex pattern.</p> <p>Note: Click Test Regex to verify if the regex expression is valid.</p>
	Delimiter	<p>Regular Expression used to delimit stream. Rules will be invoked on delimited streams.</p> <p>Note: Click Test Regex to verify if the regex expression is valid.</p> <p>Note: If the delimiter value is not matched, then the data will be processed in non-streaming mode.</p>
Error Metrics Logging		Enable if you want to log the detailed error metrics, such as, the total number of errors, the offset of where the error was encountered, the reasons why the error was encountered, the ruleset details, and so on.
	Enabled	Enable or disable error metrics to be logged in the log file.
	Log Level	Select from the following logging levels: <ul style="list-style-type: none"> • Warning • Information

Field	Sub-Field	Description
		<ul style="list-style-type: none"> Verbose <p>Note: Ensure that the log level you select is the same or part of a higher log subset that you defined in the gateway log level.</p>

The following example snippet describes the format for the Outbound Transport Settings field for NFS service:

```
{
  "filePermissions": "770",
  "createMissingDirectory": "true"
}
```

The options for the Outbound Transport Settings field are described in the following table.

Table 16-13: NFS Outbound Transport Settings

Options	Description	Default (if any)
filePermissions	Set the file permissions. Note: This setting applies only to the NFS service.	n/a
createMissingDirectory	Set to true if you want to create lock, error, and output directory automatically.	n/a

Note: Before you start using the NFS/CIFS Tunnel or Service, ensure that the *rpcbind* service is running on the NFS/CIFS server.

16.2 Profiles

A profile is a logical grouping of rules. You can group rules for a data type in a single profile. This grouping assists you in managing the rules created for the data type.

You can also refer to another profile in a RuleSet when executing a rule as a part of an action.

The following figure illustrates the fields for a Gateway Profile.

The screenshot shows a profile creation dialog box. The 'Name' field is populated with 'Fine Grained Data Protection of Account Address'. The 'Enabled' checkbox is checked. The 'Description' field is empty.

Figure 16-10: Profile creation

The following table describes the fields for the Gateway Profile.

Table 16-14: Profile fields

Field	Description
Name	Unique name for the Profile.

Field	Description
Description	Description for the Profile.
Enabled	Enable or disable the Profile.

16.3 Rules

Rules help in configuring the actions for the Cloud Gateway.

The Action fields are as seen in the following figure.

Name*	Extract Request Message Body of Account Info Add or Update
Description	
Enabled	<input checked="" type="checkbox"/>
Action*	Exit
Has to Match ⓘ	
Must Not Match ⓘ	
Scope ⓘ	Profile

Figure 16-11: Gateway Action fields

The following table describes the fields for the Gateway Action.

Table 16-15: Gateway Action Fields

Field	Description
Name	Unique name for the Profile.
Description	Description for the Profile.
Enabled	Enable or disable the Profile.
Action	Specifies the action by the Rule.

16.3.1 Actions

Actions define a rule.

The following are the different types of in the DSG:

- Error
- Exit
- Extract
- Log
- Profile Reference
- Set User Identity
- Set Context Variable
- Transform
- Dynamic Rule Injection

16.3.1.1 Exit

The Exit option acts as a terminating action and the rules are not processed further.

The fields for the Exit action are as seen in the following figure.

Name*	Extract Request Message Body of Account Info Add or Update
Description	
Enabled	<input checked="" type="checkbox"/>
Action*	Exit
Has to Match ⓘ	
Must Not Match ⓘ	
Scope ⓘ	Profile

Figure 16-12: Exit Action type-specific fields

The following table describes the fields for the Exit Action.

Table 16-16: Exit Action type-specific fields

Field	Description
Has to Match	Regular Expression for the input to match to Exit.
Must Not Match	Negative Regular expression for the above.
Scope*	Specifies the scope of the Exit Rule.

* The following are the available options for the Scope field.

- Branch: Stop processing the child rules.
- Profile: Stop processing the rules under the same profile.
- RuleSet: Stop processing all the rules.

Note: The Exit action type must always be created as a leaf node (a rule without any child nodes).

16.3.1.2 Extract

The Extract action defines the payloads supported by the DSG.

The following payloads are supported in DSG.

- Adobe Action Message Format (AMF)
- Binary
- Character Separated Values (CSV)
- Common Event Format (CEF)
- eXtensible Markup Language (XML)
- Extensible Markup Language (XML) with Tree-of-Tress (ToT)
- Fixed Width
- HTML Form Media Type (X-WWW-FORM-URLENCODED)
- HTTP Message

- JavaScript Object Notation (JSON)
- JavaScript Object Notation (JSON) with Tree-of-Tress (ToT)
- Multipart Mime
- Microsoft Office 2007 Excel Document
- Microsoft Office 2013 Document
- Adobe Portable Document Format (PDF)
- Protocol Buffer (protobuf)
- Secured File Transfer
- Amazon S3 Object
- SMTP Message
- Text
- Uniform Resource Locator
- User Defined Extraction
- ZIP Compressed File

16.3.1.2.1 Adobe Action Message Format

This payload extracts AMF format from the request and lets you define regex to control precise extraction.

The fields for Adobe Action Message Format (AMF) payload are as seen in the following figure.

Name*	COMPANY DATA Etc. > Customer Addition/Search - Customer Company Name Toke
Description	Tokenizes customer name
Enabled	<input checked="" type="checkbox"/>
Action*	Extract
Payload*	Adobe Action Message Format (AMF)
Method*	Serialize
Pattern*	

Figure 16-13: AMF payload

The properties for the AMF payload are explained in the following table.

Table 16-17: AMF payload

Field	Description
Method*	Specifies the method of extraction for AMF payloads.
Pattern	Regular Expression pattern to match and extract from string value of AMF payload

* The following options are available for the Method field.

- Serialize: Configure the AMF payload only to be exposed in learn mode. This will be useful for debugging while creating rules for learn mode.
- Serialized String Value: Configure the AMF payload as string and extract the matched Pattern.

- String Value: Configure the data using the matched Pattern. The data is not serialized ahead of the pattern matching.
- String Value by Key Name: The data is expected to come in key-value pairs. The parameters are matched using the Pattern. The value for the matched parameter is extracted.

16.3.1.2.2 Binary Payload

This payload extracts binary data from the request and lets you define regex to control precise extraction.

The fields for Binary payload are as seen in the following figure.

The screenshot shows a configuration interface for a 'Binary' payload. The fields are as follows:

- Name:** COMPANY DATA | Etc. > Customer Addition/Search - Customer Company Name Tokenization
- Description:** Tokenizes customer name
- Enabled:**
- Action***: Extract
- Payload***: Binary
- Prerequisite Match Pat...**: (empty)
- Pattern**: (empty)
- Pattern Group Id**: 0
- Encoding**: No Encoding

Figure 16-14: Binary Payload

The properties for the Binary payload are explained in the following table.

Table 16-18: Binary Payload

Field	Sub-Field	Description
Prerequisite Match Pattern		A regular expression to be searched for in the input is specified in the field.
Pattern		The regular expression pattern on which the extraction is applied is specified in this field. For example, consider if the text is "Hello World", then pattern would be "\w+".
	Pattern Group Id	The grouping number to extract from the Regular Expression Pattern. For example, for the text "Hello World", Group Id would be 0 to match characters in first group as per regex.
	Profile Name	Profile to be used to perform transform operations on the matched content.
	User Comments	Additional information related to the action performed by the group processing.
Encoding		The encoding method used while extracting binary payload.
Prefix		Prefix text to be padded before the protected value. This helps in identifying a protected text from a clear one.
Suffix		Suffix text to be padded after the protected value. The use is same as above.

Field	Sub-Field	Description
Padding Character		Characters to be added to raise the number of characters to minimum required size by the Protection method.
Minimum Input length		Number of characters that define if input is too short for the Protection method to be padded with Padding Character

The following table describes the fields for Encoding.

Table 16-19: Encoding Field

Field	Description
Codec	Select the appropriate codec based on the selection of Encoding

The following options are available for the Encoding field:

- No Encoding
- Standard
- External
- Proprietary

16.3.1.2.2.1 No Encoding

If the No Encoding option is selected, then no encoding is applied.

16.3.1.2.2.2 Standard

The Standard Encoding consists of built-in codecs of standard character encodings or mapping tables, including UTF-8, UTF-16, ASCII and more.

For more information about the complete list of encoding methods. refer to the section [Standard Encoding Method List](#).

16.3.1.2.2.3 External

When external encoding is applied, you must select a codec.

The following table describes the codecs for the External encoding.

Table 16-20: External Encoding

Codec	Description
Base64	Binary to text encoding to represent binary data in ASCII format.
HTML Encoding	Replace special characters "&", "<" and ">" to HTML-safe sequences.
JSON Escape	Escapes special JSON characters, such as quote ("") in JSON string values to make it JSON-safe sequences.
URI Encoding	RFC 2396 Uniform Resource Identifiers (URI) requires each part of URL to be quoted. It will not encode '/'.
URI Encoding Plus	It is similar to URI Encoding, except replacing '' with '+'.
XML Encoding	Escape &, <, and > in a string of data, then quote it for use as an attribute value to XML-safe sequences.
Quoted Printable	Convert to/from quoted-printable transport encoding as per RFC 1521.
SQL Escape	Performs SQL statement string escaping by replacing single quote ('') with two single quotes (''), replaces single double quote (") with two double quotes ("").



16.3.1.2.2.4 Proprietary

When proprietary encoding is selected, codecs linked are displayed.

The following table describes the codecs for the Proprietary encoding.

Table 16-21: Proprietary Fields

Codec	Description
Base128 Unicode CJK	Base128 encoding in Chinese, Japanese and Korean characters.
High ASCII	Character encodings of for eight bit or larger.

Note: The following encryption methods are not supported for the High ASCII codec and the Base128 Unicode CJK codec:

- AES-128
- AES-256
- 3DES
- CUSP AES-128
- CUSP AES-256
- CUSP 3DES
- FPE NIST 800-38G Unicode (Basic Latin and Latin-1 Supplement Alpha)
- FPE NIST 800-38G Unicode (Basic Latin and Latin-1 Supplement Alpha-Numeric)

Note: The following tokenization data types are not supported for the High ASCII codec and the Base128 Unicode CJK codec:

- Binary
- Printable

Note: The input data for the Base128 Unicode CJK and High ASCII codecs must contain only ASCII characters. For example, if input data consisting of non-english characters is tokenized using the Alpha tokenization, then the Alpha tokenization treats the non-english characters as a delimiter and the tokenized output will include the non-english characters. As a result, the protection or unprotection operation will fail.

Note: In versions prior to DSG 3.0.0.0, if the input data consists of non-english characters and is not configured to get encoded, even then the data gets encoded. From DSG 3.0.0.0, this behavior is corrected, where only the extracted data is encoded.

16.3.1.2.3 Common Event Format (CEF)

If you want to protect fields that are part of a CEF log file, you can use the CEF payload to extract the required fields.

Payload*

Line Separation

Fields

Field Name*

Profile Name

User Comments

NOTE: Recursion rule reference detected. Recursions are allowed with the limit of max depth configuration (default to 10)

Figure 16-15: CEF Payload

The properties for the Common Event Format (CEF) payload are explained in the following table.

Table 16-22: Common Event Format (CEF) Payload

Properties	Sub-Field	Description
Line Separator		Regex pattern to identify field separation.
Fields		CEF names and profile references must be selected.
	Field Name	Comma separated list of CEF key names that need to be transformed (protected or unprotected).
	Profile Name	Profile to be used to perform transform operations on the matched content.
	User Comments	Additional information related to the action performed by the column processing.

16.3.1.2.4 Date Time Format

The Datetime format payload is used to convert custom datetime formats, which are not supported by tokenization datetime or date data element, to a supported format that can be processed by DSG.

Consider an example, where you provide a time format, such as DD/MM/YYYY HH:MM:SS as an input to an Extract rule with the Datetime payload. The given format is not supported by the datetime or date tokenization data element. The Extract rule converts the format to an acceptable format, a transform rule protects the datetime. The Datetime payload converts the protected value to the input format and returns this value to the user.

When you request DSG to unprotect the protected datetime value, an extract rule identifies the protected datetime value, a subsequent transform rule unprotects the value and returns the original datetime format, which is DD/MM/YYYY HH:MM:SS.

Caution: Ensure that the input sent to the extract rule for Date Time extraction is exactly in the same input format as configured in the rule. If you are unsure of the input that might be sent to the extract rule, then ensure that before you rollout for production, Ruleset configuration is thoroughly checked.

The following figure illustrates the Date Time format payload fields.

Action*	Extract
Payload*	Date Time Format
Input Date Time Format* ⓘ	DD/MM/YYYY HH:MM:SS
Data Element Date Time Format* ⓘ	YYYY-MM-DD HH:MM:SS MMM
Mode of Operation* ⓘ	Unprotect
Distinguishable Date ⓘ	<input type="checkbox"/>

Figure 16-16: Date Time Format codec

Before you begin:

Ensure that the following pre-requisites are completed:

- The datetime data element defined in the policy on ESA is used to perform protect or unprotect operation.

The following table describes the fields for Datetime codec.

Table 16-23: Datetime codec fields

Field	Description
Input Date Time Format	Format in which the input is provided to DSG. Note: This field accepts numeric values only in the input request sent to DSG.
Data Element Date Time Format	Format to which input must be converted. Note: Ensure that the Transform rule that follows the Extract rule uses the same data element that is used to configure the Date Time Format codec.
Mode of Operation	Data security operation that needs to be performed. You can select Protect or Unprotect. Note: The mode of operation must be same as the data security operation that you want to perform in the Transform rule.
DistinguishableDate*	Select this checkbox if the data element used to protect the date time is included this setting.

*These fields appear only when Unprotect is selected as Mode of Operation.

16.3.1.2.5 CSV Payload

This payload extracts CSV format from the request and lets you define regex to control precise extraction.

With the Row and Column Index, you can now define how the column positions can be calculated. For example, consider a CSV input as provided in the following snippet, where the first column begins at the 0th field in the row, that are padded with commas until the next column, ex5 begins. This applies when the indexing is 0-based. If you choose to use 1-based indexing, the first column begins at 1 and subsequent fields are 2, 3, and so on. Based on these definitions, you can define the rule and its properties.

```
first, ex5, last, pick, ex6, city, ex1, ex2
John, www, Smith, mister, www, stamford, 333, 444
Adam, www, Martin, mister, www, fairfield, 333, 444
```

Note: It is recommended to use the External Base64 encoding method in the Transform action type for the CSV codec. If the Standard Base64 method is used, then additional newline feeds are generated in the output.

Note:

The CSV implementation in the DSG does not support the following:

- The fields contain line breaks, double quotes, and commas enclosed in double quotes.
- If double quotes are used to enclose fields, then the double quotes appearing inside a field are escaped by preceding them with another double quote.

The fields for the CSV payload are as seen in the following figure.



The screenshot shows the configuration interface for a CSV payload. Key settings include:

- Payload***: Character Separated Values (CSV)
- Line Separator**: \r\n
- Skip Lines Matching Pattern**: (empty)
- Row and Column Index**: 0
- Header Line Number**: 0
- Data Starts at Line**: 1
- Column Separator**: ,
- Columns**: (empty)
- Column Name/Index***: 1
- Profile Name**: REST API Examples > Protect Rule
NOTE: Recursion rule reference detected. Recursions are allowed with the limit of max depth configuration (default to 10)
- User Comments**: (empty)
- Text Qualifier**: (empty)
- pattern**: (empty)

Figure 16-17: CSV payload

Caution: If the CSV input includes NON-ASCII or Unicode data, then the *Binary* extract rule must be used before using the *CSV* extract rule.

Note:

If the CSV input file includes non-printable special characters, then to transform the data successfully, the user must add the *csv-bytes-parsing* parameter in the *features.json* file.

To add the parameter in the *features.json* file, perform the following steps.

1. Login to the ESA CLI Manager.
2. Navigate to **Administration > OS Console**.
3. Navigate to the **/opt/protegility/alliance/config** directory.
4. Open the *features.json* file for editing.
5. Add the *csv-bytes-parsing* parameter in the *features.json* file. The *csv-bytes-parsing* parameter must be added in the following format:

```
{
  "features": [
    "csv-bytes-parsing"
  ]
}
```

The properties for the CSV payload are explained in the following table.

Table 16-24: CSV Payload

Properties	Sub-Field	Description	Additional Information
Line Separator		Separator that defines where a new line begins.	

Properties	Sub-Field	Description	Additional Information
Skip Lines Matching Pattern		Regex pattern that defines the lines that need to be skipped.	<p>For example, consider the following lines in the file:</p> <pre>User, Admin, Full Access, Viewer Partial Access, User, Viewer, Admin No Access, Viewer, User, Root No Acess, Partial Access, Root, Admin</pre> <ul style="list-style-type: none"> If you configure the regex as <code>.*?User</code>, the lines 1, 2, and 3 will be skipped. If you configure the regex as <code>User</code>, the first line will be skipped and the remaining lines will be processed.
Preserve Number of Columns		<p>Select to check if the number of columns are equal to the column headers in a CSV file.</p> <p>If there is a mismatch between the actual number of columns and the number of column headers, then the rule stops processing further and an error appears in the log.</p> <p>If you clear this check box and a mismatch is detected, then the rule still continues to process the data. A warning appears in the log.</p>	
Row and Column Index		Select 0 if row and column counting begins at 0 or 1 if it begins at 1.	0
Header Line Number		Line number with column headers.	<ul style="list-style-type: none"> -1 – Row and Column Index is 0 0 – Row and Column Index is 1
Data Starts at Line		Line number from which the data begins.	Value calculated as Header Line Number +1
Column Separator		Value by which the columns are separated.	
Columns		List of columns to be extracted and for which values action is to be applied. For example, consider a .csv file with multiple columns such as SSN, Name, etc that need to be processed.	
	Column Name/Index	<p>Column Name or index number of the column that will be processed. For example, if the name of the 1 column is “Name”, the value in Column Name/Index would be either 1 or Name.</p> <p>For example, with Row and Column Index defined as 0, if</p>	

Properties	Sub-Field	Description	Additional Information
		the name of the 1st column is “Name”, the value in Column Name/Index would be either 0 or Name.	
	Profile Name	Profile to be used to perform transform operations on the matched content.	
	User Comments	Additional information related to the action performed by the column processing.	
Text Qualifier		Pattern that allows cells to be combined.	
Pattern		Pattern that applies to the cells, after the lines and columns have been separated.	
Advanced Settings		<p>Define the quote handling for quotes in the CSV records.</p> <ul style="list-style-type: none"> • Set to <code>{"quoteHandlingMode" : "DEFAULT"}</code> to correct unbalanced quotes in records, such as, single quotes, in the delimited CSV input file during data processing. For example, if the CSV includes unbalanced quotes, such as, <code>' ,03/11/2020</code> or <code>" ,13/08/2020</code> and the <i>Default Mode</i> is enabled, then during data processing, the DSG will correct the unbalanced quotes. The DSG will change the unbalanced quotes to <code>' ',03/11/2020</code> or <code>" ",13/08/2020</code> respectively. • Set to <code>{"quoteHandlingMode" : "PASSIVE"}</code> to retain unbalanced quotes in records, such as single quotes, in the delimited CSV input file during data processing. For example, if the CSV includes a unbalanced quotes, such as, <code>' ,03/11/2020</code> or <code>" ,13/08/2020</code> and the <i>Passive Mode</i> is enabled, then during data processing, the DSG will retain the unbalanced quotes. 	

Properties	Sub-Field	Description	Additional Information
		<ul style="list-style-type: none"> Set to <code>{"quoteHandlingMode" : "DISABLE"}</code> to protect the quotes in records, such as single quotes, in the delimited CSV input file during data processing. <p>For example, if the CSV includes quotes, such as, '<code>abcdefg</code>' or "<code>abcdefg</code>" and the <i>Disable Mode</i> is enabled, then during data processing, the DSG will protect the quotes in the input file.</p>	

16.3.1.2.6 XML Payload

This payload extracts the XML format content from the request and lets you extract the exact XML element value with it.

The fields for the XML payload are as seen in the following figure.

Payload* eXtensible Markup Language (XML)

XPath List* /class_list/student/name

Advance XML Parser options {"remove_blank_text": true}

Figure 16-18: XML payload

The properties for the XML payload are explained in the following table.

Table 16-25: XML Payload

Properties	Description
XPath List	<p>The XML element value to be extracted is specified in this field.</p> <p>Note: Ensure that you enter the XPath by following proper syntax for extracting the XML element value. If you enter incorrect syntax, then the service which has this XML payload definition in the rule fails to load and process the request.</p>
Advance XML Parser options*	<p>Configure advanced parsing parameter options for the XML payload. This field accepts parsing options in the JSON format. The parsing options are of the Boolean data type.</p> <p>For example, the parsing parameter, <code>remove_comments</code>, accepts the values as <code>true</code> or <code>false</code>.</p>

* The *Advance XML Parser options* field provides the following parsing parameters that can be configured.

Table 16-26: Additional XML Parser options

Options	Description	Default
remove_blank_text	Boolean value used to remove the whitespaces for indentation in the XML payload.	False
remove_comments	Boolean value used to remove comments from the XML payload. In the XML format, comments are entered in the <!-- --> tag.	False
remove_pis	Boolean value used to remove Processing Instructions (pi) from the XML payload. In the XML format, processing instructions are entered in the <? -- ?> tag.	False
strip_cdata	Boolean value used to replace content in the cdata, Character data, or tag by normal text content.	True
resolve_entities	Boolean value used to replace the entity value by their textual data value.	False
no_network	Boolean value used to prevent network access while searching for external documents.	True
ns_clean	Boolean value used to remove redundant namespace declarations.	False

Consider the following example to understand the *Advance XML Parser* options available in the XML codec. In this example, a request is sent from a client to remove the whitespaces between the XML tags from a sample XML payload in the message body of the HTTP/REST request. The following Ruleset is created for this example.

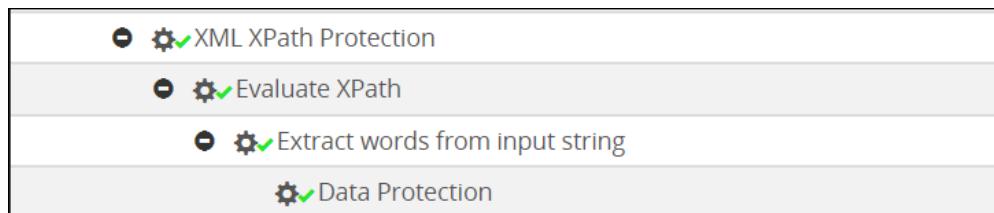


Figure 16-19: Ruleset structure

1. Create an extract rule for the HTTP message payload using the default RuleSet template defined under the REST API service.
Consider the following sample XML payload in the HTTP message body.

```

<?xml version = "1.0" encoding = "ASCII" ?>
<class_list>
    <!--Students grades are uploaded by months-->
    <student>
        <name>John Doe</name>
        <grade>A</grade>
    </student>
</class_list>
  
```

In the example, a lot of white spaces are used for indentation. The payload contain spaces, carriage returns, and line feeds between the `<class_list>`, `<student>`, and `<name>` XML tags.

2. The extract rule for extracting the HTTP message body is as seen in the following figure.

The screenshot shows the configuration interface for an Extract rule. The fields include:

- Name***: XML XPath Protection
- Description**: Rule hierarchy protects values contained in XML tags based on XPath. This particular rule node extracts HTTP POST message body as raw bytes.
- Enabled**: Checked
- Action***: Extract
- Payload***: HTTP Message
- HTTP Message Type***: Request
- HTTP Method**: POST
- URI**: /protect/xml/xpath
- Request Headers**
- Message Body**
- Require Client Certificate**: Unchecked
- Authentication**: None
- Target Object**: Message Body

Figure 16-20: Extract rule fields

- Under the Extract rule, create another child rule to extract the XML payload from the HTTP Message.

In this child rule, provide `/class_list/student/name` to parse the XML payload in the **XPath List** field and set the `remove_blank_text` parameter to `true` in the **Advance XML Parser options** field in the JSON format.

The screenshot shows the configuration interface for a child Extract rule. The fields include:

- Name***: Evaluate XPath
- Description**: Example input XML for this rule:
<Person>
- Enabled**: Checked
- Action***: Extract
- Payload***: eXtensible Markup Language (XML)
- XPath List***: /class_list/student/name
- Advance XML Parser options**: {"remove_blank_text": true}

Figure 16-21: Child extract rule fields

- Under this extract rule, create another child rule to extract the sensitive data between the `<name>` and `</name>` tags. The fields for this child extract rule are as seen in the following figure.

Name*: Extract words from input string

Description:

Enabled:

Action*: Extract

Payload*: Text

Prerequisite Match Pattern ⓘ:

Pattern ⓘ: \w+

Pattern Group Ids ⓘ:

Encoding ⓘ: No Encoding

Edit

Figure 16-22: Extracting sensitive data between XML tags

- Under the extract rule, create a transform rule to protect the sensitive data between the <name> and the <name> tags using Regex Replace with a pattern **xxxxx**. The fields for the transform rule are as seen in the following figure.

Name*: Data Protection

Description:

Enabled:

Action*: Transform

Method*: Regular Expression Replace

Replace Pattern*:

Match Pattern*: (.*)

Replace Value: xxxx

Edit

Figure 16-23: Transform rule fields

- Click **Deploy** or **Deploy to Node Groups** to apply the configuration changes.

When a request is sent to the configured URI, the DSG processes the request and the following response appears with the whitespaces removed from the XML payload. In addition, the sensitive data between the `<name>` and the `</name>` tags is protected.

```
<?xml version='1.0' encoding='ASCII'?>
<class_list><!--Students grades are uploaded by months--><student><name>xxxxxx xxxx</name><grade>A</grade></student></class_list>
```

16.3.1.2.7 XML with Tree-of-Trees (ToT) Payload

The XML with Tree-of-Trees (ToT) codec extracts the XML element defined in the **XPath** field. The XML with ToT codec allows you to process the multiple XML elements in an extract rule.

The fields for the *XML with ToT* payload is as seen in the following figure.

Payload*	XML with Tree-of-Trees
XML Paths with Profile Reference*	
<input checked="" type="checkbox"/> XPath*	<input type="text"/>
Profile Name	<input type="text"/>
User Comments	<input type="text"/>
<input checked="" type="checkbox"/> XPath*	<input type="text"/>
Profile Name	<input type="text"/>
User Comments	<input type="text"/>
Advance XML Parser o...	

Figure 16-24: XML with ToT payload

To understand the XML with ToT payload, consider the following example where the student details, such as, name, age, subject, and gender can be sent as a part of the request. In this example, the *XML with ToT* rule extracts and protects the name and the age element.

```
<?xml version='1.0' encoding='UTF-8'?>
<students>
  <student>
    <name>Rick Grimes</name>
    <age>35</age>
    <subject>Maths</subject>
    <gender>Male</gender>
  </student>
  <student>
    <name>Daryl Dixon </name>
    <age>33</age>
    <subject>Science</subject>
    <gender>Male</gender>
  </student>
  <student>
    <name>Maggie</name>
    <age>36</age>
    <subject>Arts</subject>
    <gender>Female</gender>
  </student>
</students>
```

The following figure illustrates one extraction rule for multiple XML elements.

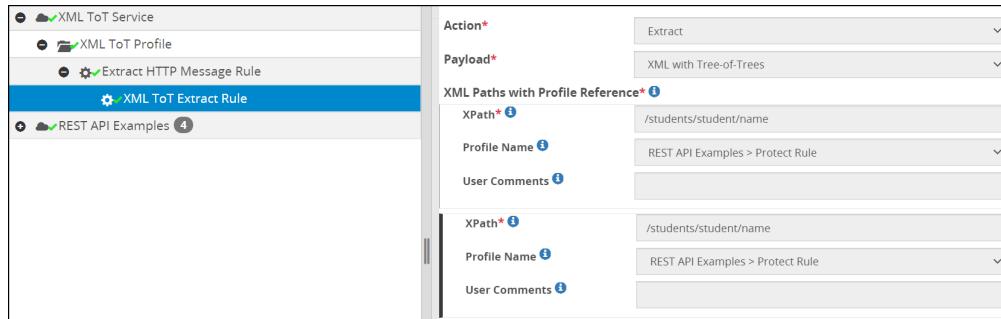


Figure 16-25: XML ToT extraction and transformation rule

In the [Figure 14-25](#), the **XML ToT Extract rule** extracts the two different XML elements, name and age. The `/students/student/name` path extracts the name element and protect it with the transform rule. Similarly, the `/students/student/age` path extracts the age element and protect it with the transform rule. Same data elements are used to protect both the XML elements. You can use different data elements to transform XML elements as per your requirements. It is recommended to use the **Profile reference**, from the drop-down that appears on the **Profile Name** field. This helps to process the extraction and transformation in one rule. In addition, it reduces the transform overhead of defining one element at a time for the same XML file. If the **Profile Name** field is left empty then the extracted value is passed to the child rule for transformation.

For more information about profile referencing, refer to the section [Profile Reference](#).

The properties for the **XML with ToT payload** are explained in the following table.

Table 16-27: XML with ToT Payload

Properties	Subfield	Description
XPaths with Profile Reference		<p>Define the required XPath and Profile reference.</p> <p>Note: Ensure that you enter the XPath by following the required syntax for extracting the XML element value. For example, in the Figure 14-25, the <code>/students/student/name</code> path is defined for the name element, ensure to follow the same syntax to extract the XML element. If you enter an incorrect syntax, then the defined rule is disabled.</p>
	XPath	Define the required XML element.
	Profile Name	Select the required transform rule.
	User Comments	Add additional information for the action performed if required.
Advance XML Parser options		<p>Configure advanced parsing parameter options for the XML payload. This field accepts parsing options in the JSON format. The parsing options are of the <i>Boolean</i> data type.</p> <p>For example, the parsing parameter, <code>remove_comments</code>, accepts the values as <code>true</code> or <code>false</code>.</p> <p>Note:</p> <p>The Advance XML Parser options that apply to the XML codec also apply to the XML with ToT codec.</p>

Properties	Subfield	Description
		For more information about the additional XML Parser, refer to the Table 14-26: Advance XML Parser .

16.3.1.2.8 Fixed Width

In scenarios where the input data is sent to DSG in a fixed width format, the Fixed Width codec is used. In a fixed width input, the data columns are specified in terms of exact column start character offset and fixed column width in terms of number of characters that define column width.

For example, consider a fixed width input as provided in the following snippet. The Name column begins at the 0th character in a row, has a fixed width of 20 characters and is padded with spaces until the next column Number begins. The Number column begins at the 20th character in a row and has a fixed width of 12 characters.

With the Row and Column Index, you can now define how the column positions can be calculated. If you choose to use 1-based indexing, the Name column begins at 1 and for fixed width of 20 characters, subsequent column will begin at the 21st character. While, if you use 0-based indexing, the Name column begins at 0 and for fixed width of 20 characters, subsequent column will begin at the 20th character. Based on these definitions, you can define the rule and its properties.

Name	Number
John Smith	418-Y11-4111
Mary Hartford	319-Z19-4341
Evan Nolan	465-R45-4567

The fields for the Fixed Width payload are as seen in the following figure.

Figure 16-26: Fixed Width Payload

Note:

If the input file includes non-printable special characters, then to transform the data successfully, the user must add the *fw-bytes-parsing* parameter in the *features.json* file.

To add the parameter in the *features.json* file, perform the following steps.

1. Login to the ESA CLI Manager.
2. Navigate to **Administration > OS Console**.
3. Navigate to the `/opt/protegility/alliance/config` directory.
4. Open the `features.json` file for editing.
5. Add the `fw-bytes-parsing` parameter in the `features.json` file. The `fw-bytes-parsing` parameter must be added in the following format:

```
{
  "features": [
    "fw-bytes-parsing"
  ]
}
```

The properties for the Fixed Width payload are explained in the following table.

Table 16-28: Fixed Width Payload

Properties	Sub-Field	Description
Line Separator		Separator that defines where a new line begins.
Skip Lines Matching Pattern	Regex pattern that defines the lines that need be skipped.	<p>For example, consider the following lines in the file:</p> <pre>User, Admin, Full Access, Viewer Partial Access, User, Viewer, Admin No Access, Viewer, User, Root No Acess, Partial Access, Root, Admin</pre> <ul style="list-style-type: none"> • If you configure the regex as <code>.*?User</code>, the lines 1, 2, and 3 will be skipped. • If you configure the regex as <code>User</code>, the first line will be skipped and the remaining lines will be processed.
Preserve Input Length		<p>Select to perform a check for the input and output length. If a mismatch is detected, then the rule stops processing further and an error appears in the log.</p> <p>If you clear this check box and a mismatch is detected, the rule still continue processing the data. A warning appears in the log.</p>
Row and Column Index		Select 0 if row and column counting begins at 0 or 1 if it begins at 1.
Data Starts at Line		Line number from which the data begins.
Fixed Width Columns		
	Column Position	<p>Column position where the data begins. For example, if you are protecting the first column with 20 characters fixed width, then the value in this field will be 0.</p> <p>This value differs based on the Row and Column Index defined. For example, if you choose to use 0-based indexing, then the first column begins at 0, and the value in this field will be 0.</p>

Properties	Sub-Field	Description
	Column Width	The fixed width of the column that must be protected. For example, if you are protecting the first column with 20 characters fixed width, then the value in this field will be 20.
	Profile Name	Profile to be used to perform transform operations on the matched content. Note: Ensure that the data element used to perform the transfer operation is of Length Preserving type.
	User Comments	Additional information related to the action performed by the column processing.

16.3.1.2.9 HTML Form Media Payload (X-WWW-FORM-URLENCODED)

This payload extracts HTML form media format from the request and lets you define regex to control precise extraction.

The fields for the HTML Form Media Payload (X-WWW-FORM-URLENCODED) payload are as seen in the following figure.

The screenshot shows the configuration interface for an HTML Form Media Type payload. It includes fields for Name, Value, Target Object, Encoding Mode, and Encoding Reserve Characters, each with dropdown menus and 'Test Regex' buttons.

Figure 16-27: HTML Form Media Type payload

The properties for the X-WWW-FORM-URLENCODED payload are explained in the following table.

Table 16-29: X-WWW-FORM-URLENCODED Payload

Properties	Description
Name	The regular expression to match the parameter name is specified in this field.
Value	The value to be extracted is specified in this field.
Target Object	The parameter object to be extracted is specified in this field.
Encoding Mode	Encoding mode that will be used for URI encoding handling.
Encoding Reserve Characters	Characters beyond uppercase and lowercase alphabets, underscore, dot, and hyphen.

16.3.1.2.10 HTTP Message Payload

This payload extracts HTTP message format from the request and lets you define regex to control precise extraction.

The following figure illustrates the HTTP Message payload fields.

The screenshot shows the configuration interface for an 'HTTP Message' payload. The fields are as follows:

- Payload***: HTTP Message
- HTTP Message Type***: Request
- HTTP Method**: (empty input field)
- URI**: (empty input field) with a **Test Regex** button
- Request Headers**: (empty input field) with a **+ Add** button
- Message Body**: (empty input field) with a **Test Regex** button
- Require Client Certifica...**: Unchecked checkbox
- Authentication**: None
- Target Object**: Message Body

Figure 16-28: HTTP Message payload

The properties for the HTTP Message payload are explained in the following table.

Table 16-30: HTTP Message Payload

Properties		Description
HTTP Message Type		Type of HTTP Message to be matched.
Method		The value to be extracted is specified in this field.
Request URI		The regular expression to be matched with the request URI is specified in this field.
Request Headers		The list of name and value as regular expression to be matched with the request headers is specified in this field.
Message Body		The parameter object to be extracted is specified in this field.
Require Client Certificate		If checked, the client must present a certificate for authentication. If no certificate is provided, a 401 or 403 response appears.
Authentication		Authentication rule required for the rule to execute. Authentication mode can be none or basic authentication.
Target Object		<p>The target message body to be extracted is specified in this field. The following Target Object options are available:</p> <ul style="list-style-type: none"> • Message Body • Cookie • Message Header • Message Headers • Client Certificate* • Uniform Resource Locator (URL) <p>Client Certificate* - The following fields are displayed if the Client Certificate option is selected in the Target Object drop down menu:</p> <ul style="list-style-type: none"> • Attribute • Value • Target Object

Properties		Description
	Attribute	<p>The client certificate attributes to be extracted are specified in this field. The following attribute options are available:</p> <ul style="list-style-type: none"> • issuer • notAfter • notBefore • serialNumber • subject\$ • subjectAltName • version • crlDistributionPoints • caIssuers • OCSP
	Value	<p>Regular expression to identify the client certificate attributes to be extracted. The default value is (.*)</p>
	Target Object	<p>The value or the attribute of the client certificate to be extracted is specified in this field. The following Target Object options are available:</p> <ul style="list-style-type: none"> • Value • Attribute

16.3.1.2.11 JSON Payload

This codec extracts the JSON element from the JSON request as per the JSONPATH defined.

Consider the following sample input that will be processed using the JSON codec to extract a unique JSON element:

```
{
  "entities": [
    {
      "entity_type": "CostCenter",
      "properties": {
        "Id": "10097",
        "LastUpdateTime": 1455383881190,
        "Currency": "USD",
        "ApproveThresholdAmount": 100000,
        "AccountingCode": "5555",
        "CostCenterAttachments": "[ ]"
      }
    }
  ],
  "operation": "UPDATE"
}
```

In the *Extract* rule, assuming that the *AccountingCode* needs to be protected, the JSONPath that will be set is *entities[*].properties.AccountingCode*. Based on the input JSON structure, the JSONPath value differs.

The following figure illustrates the JSON payload fields.

The screenshot shows the configuration interface for a JSON payload. The 'Payload*' dropdown is set to 'Javascript Object Notation (JSON)'. The 'Allow Empty String' checkbox is checked. The 'JSONPath*' input field contains 'entities[*].properties.AccountingCode'. The 'Preserve Element Order' checkbox is unchecked. The 'Fail Transaction' checkbox is checked. The 'Minimize Output' checkbox is unchecked. The 'Process Mode' dropdown is set to 'Complex - Stringify'. The 'Rule Advanced Settings' section is collapsed.

Figure 16-29: JSON payload

The properties for the JSON payload are explained in the following tab

Table 16-31: JSON payload

Properties	Sub-Field	Description
JSONPath		<p>This JSON element value to be extracted is specified in the JSON path.</p> <p>Note: Ensure that you enter the JSONPath by following proper syntax for extracting the JSON element value. If you enter incorrect syntax, then the service which has this JSON payload definition in the rule fails to load and process the request.</p>
Allow Empty String		<p>Enable to pass values that are defined as only whitespaces, such as, <i>value: " "</i>, that are part of the JSON payload and continue processing of the sequential rules. If this check box is disabled, then the <i>Extract</i> rule does not process values that are defined as only whitespaces.</p>
Preserve Element Order		<p>Select to preserve the order of key-value pairs in the JSON response.</p>
Fail Transaction		<p>Select to fail transaction when an error is encountered during tokenization. The error might occur due to use of incorrect token element to protect input data. For example, when handling integer input data, the accurate token element would be an integer token element.</p> <p>The DSG uses tokenization logic to perform data protection. This logic can work to its optimum only if the correct token element is used to protect the input data. If you fail to perform careful analysis of your input data and identify the accurate token element that</p>

Properties	Sub-Field	Description
		<p>must be used, then it will result in issues when data is protected using the tokenization logic. To avoid this issue, it is recommended that before defining rules, analyze the input data and identify the accurate token element to be used to protect the data.</p> <p>For more information about identifying the token element that will best suit the input data, refer to Protection Method Reference Guide 9.1.0.0.</p>
Minimize Output		<p>Select to minify the JSON response. The JSON response is displayed in a compact form as opposed to the indented JSON response that the DSG sends.</p> <div style="background-color: #e0f2f1; padding: 10px;"> <p>Note:</p> <p>It is recommended that this option is selected when the JSON input includes deeply nested key-value pairs.</p> </div>
Process Mode		<p>Select to parse JSON data types. This field includes the following three options:</p> <ul style="list-style-type: none"> • Simple - Primitive • Complex - Stringify • Complex - Recurse
	Complex - Stringify	<p>Select to process the complex JSON data type, such as, arrays and objects, to string values and serializing to JSON values before being passed to the child rule. This option is displayed by default.</p>
	Simple - Primitive	<p>Select to process primitive data types, namely, string, int, float, and boolean. It does not support the processing of complex data types, such as, arrays and objects when it matches the JSON data type; the processing fails and an error message is displayed.</p>
	Complex - Recurse	<p>Select to process the complex JSON data type and iterate through the JSON array or object recursively.</p>

The following table describes the additional configuration option for the Recurse mode.

Table 16-32: Additional Recurse mode configuration option

Options	Description	Default
recurseMaxDepth	<p>Maximum recursion depth that can be set for iterating matched arrays or objects.</p> <div style="background-color: #ffd700; padding: 5px;"> <p>Caution: This parameter comes in effect only when the Complex - Recurse mode is selected. It is not supported for the</p> </div>	25

Options	Description	Default
	Complex - Stringify and the Simple - Primitive modes.	

16.3.1.2.11.1 JSONPath Examples

This section provides guidance on the type of JSONPath expressions that DSG understands. This guidance must be considered before you define the acceptable JSONPath to be extracted when using the JSON codec.

The DSG supports the following operators.

Caution:

The `$` operator is not supported.

Table 16-33: Operators supported in DSG

Operator	Description	Example
*	Wildcard to select all elements in scope.	<code>foo.*.baz</code> <code>["foo"][*]["baz"]</code>
..	Skip any number of elements in path.	<code>foo..baz</code>
[]	Access arrays or names with spaces in them.	<code>["foo"]["bar"]["baz"]</code> <code>array[-1].attr</code> <code>[3]</code>
array[1:-1:2]	Slicing arrays.	<code>array[1:-1:2]</code>
=, >, <, >=, <= and !=	Filter using these elements.	<code>foo(bar.baz=true)</code> <code>foo.bar(baz>0).baz</code> <code>foo(bar="yawn").bar</code>

To understand the JSONPath, consider the following example JSON. The subsequent table provides JSONPath examples that can be used with the example JSON.

```
{
  "store": {
    "book": [
      {
        "category": "reference",
        "author": "Nigel Rees",
        "title": "Sayings of the Century",
        "price": 8.95
      },
      {
        "category": "fiction",
        "author": "J. R. R. Rowling",
        "title": "Harry Potter and Chamber of Secrets",
        "isbn": "0-395-12345-8",
        "price": 29.99
      },
      {
        "category": "fiction",
        "author": "J. R. R. Tolkien",
        "title": "The Lord of the Rings",
        "isbn": "0-395-19395-8",
        "price": 22.99
      }
    ]
  }
}
```



```

    "category": "fiction",
    "author": "Arthur Conan Doyle",
    "title": "Sherlock Homes",
    "isbn": "0-795-19395-8",
    "price": 9
  }
]
}
}

```

The following table provides the JSONPath examples based on the JSON example.

Table 16-34: JSONPath examples

JSONPath	Description	Notes
<code>store..title</code>	All titles are displayed.	The given JSONPath examples are different in construct but provide the same result.
<code>store.book[*].title</code>		
<code>store.book..title</code>		
<code>["store"]["book"][*] ["title"]</code>		
<code>store.book[0].title</code>	The first title is displayed.	The given JSONPath examples are different in construct but provide the same result.
<code>["store"]["book"][0] ["title"]</code>		
<code>store.book[1:-1].title</code>	All titles except first and last title are displayed.	The given JSONPath examples are different in construct but provide the same result.
<code>["store"]["book"][1:-1] ["title"]</code>		
<code>["store"]["book"](<code>price</code>>=9) ["title"]</code>	All titles with book price greater than or equal to 9 or 9.0.	
<code>["store"]["book"](<code>price</code>>9) ["title"]</code>	All titles with book price greater than 9 or 9.0.	
<code>["store"]["book"](<code>price</code><9) ["title"]</code>	All titles with book price less than 9 or 9.0.	
<code>["store"]["book"](<code>price</code><=9) ["title"]</code>	All titles with book price less than or equal to 9 or 9.0.	

16.3.1.2.12 JSON with Tree-of-Trees (ToT)

This section provides an overview of the JSON with Tree-of-Trees (ToT) payload. The JSON ToT payload allows you to use the advantages offered by [Tree of Trees](#) to extract the JSON payload from the request and provide protection according to the data element defined. [Profile Reference](#) can also be used to process different elements of the JSON.

The following figure illustrates the JSON ToT fields.

Payload* JSON with Tree-of-Trees

Allow Empty String

JSON Paths with Profile Reference*

JSON Path* <input checked="" type="checkbox"/>	entities[*].properties AccountingCode
Profile Name <input type="checkbox"/>	
User Comments <input type="checkbox"/>	
Process Mode <input type="checkbox"/>	Complex - Stringify

Preserve Element Order

Fail Transaction

Minimize Output

Rule Advanced Settings

Figure 16-30: JSON with Tree-of-Trees

The properties for the JSON ToT payload are explained in the following table:

Table 16-35: JSON with Tree-of-Trees payload

Properties	Sub-Field		Description
Allow Empty String			Enable to pass values that are defined as only whitespaces, such as <code>value: " "</code> , that are part of the JSON payload and continue processing of the sequential rules. If this check box is disabled, then the Extract rule does not process values that are defined as only whitespaces.
JSON Paths with Profile Reference			JSON path and profile references must be selected.
	JSON Path		JSON path representing the JSON field targeted for extraction.
	Profile Name		Profile to be used to perform transform operations on the matched content.
	User Comments		Additional information related to the action performed by the group processing.
	Process Mode		Select to parse JSON data types. This field includes the following three options: <ul style="list-style-type: none"> • Simple - Primitive • Complex - Stringify • Complex - Recurse
		Complex - Stringify	Select to process the complex JSON data type, such as, arrays and objects, to string values and serializing to JSON values before being passed to the child

Properties	Sub-Field		Description
			rule. This option is displayed by default.
		Simple - Primitive	Select to process primitive data types, namely, string, int, float, and boolean. It does not support the processing of complex data types, such as, arrays and objects when it matches the JSON data type; the processing fails and an error message is displayed.
		Complex - Recurse	Select to process the complex JSON data type and iterate through the JSON array or object recursively.
Preserve Element Order			Select to preserve the order of key-value pairs in the JSON response. This option is selected by default when you create the JSON ToT rule.
Fail Transaction			<p>Select to fail transaction when an error is encountered during tokenization. The error might occur due to use of incorrect token element to protect input data. For example, when handling integer input data, the accurate token element would be an integer token element.</p> <p>The DSG uses tokenization logic to perform data protection. This logic can work to its optimum only if the correct token element is used to protect the input data. If you fail to perform careful analysis of your input data and identify the accurate token element that must be used, then it will result in issues when data is protected using the tokenization logic. To avoid this issue, it is recommended that before defining rules, analyze the input data and identify the accurate token element to be used to protect the data.</p> <p>This option is selected by default when you create the JSON ToT rule.</p> <p>For more information about identifying the token element that will best suit the input data, refer to Protection Method Reference Guide 9.1.0.0.</p>
Minimize Output			Select to minify the JSON response. The JSON response is

Properties	Sub-Field		Description
			<p>displayed in a compact form as opposed to the indented JSON response that the DSG sends.</p> <p>This option is deselected by default when you create the JSON ToT rule.</p> <p>Note: It is recommended that this option is selected when the JSON input includes deeply nested key-value pairs.</p>

The following table describes the additional configuration option for the Recurse mode.

Table 16-36: Additional Recurse mode configuration option

Options	Description	Default
recurseMaxDepth	<p>Maximum recursion depth that can be set for iterating matched arrays or objects.</p> <p>Caution: This parameter comes in effect only when the Complex - Recurse mode is selected. It is not supported for the Complex - Stringify and the Simple - Primitive modes.</p>	25

16.3.1.2.13 Multipart Mime Payload

This payload extracts mime payload from the request and lets you define regex to control precise extraction.

The following figure illustrates the Multipart Mime payload.



Figure 16-31: Multipart Mime payload

The properties for the Multipart Mime payload are explained in the following table.

Table 16-37: Multipart Mime payload

Properties	Description
Headers	Name-Value pair of the headers to be intercepted.
Message Body	Intercept the message matching the regular expression.
Target Object	Target message to be extracted.

16.3.1.2.14 Microsoft Office Documents

This payload extracts Microsoft Office documents from the request and lets you define regex to control precise extraction. The following figure illustrates the MS Office payload fields.

Figure 16-32: Microsoft Office Documents

The properties for the Microsoft Office documents payload are explained in the following table.

Table 16-38: Microsoft Office Documents

Properties	Sub-Field	Description
Pattern		The regular expression pattern on which the extraction is applied is specified in this field. For example, consider if the text is “Hello World”, then pattern would be “\w+”.
	Pattern Group Id	The grouping number to extract from the Regular Expression Pattern. For example, for the text “Hello World”, Group Id would be 0 to match characters in first group as per regex.
	Profile Name	Profile to be used to perform transform operations on the matched content.
	User Comments	Additional information related to the action performed by the group processing.
Length Preservation		Data transformation output is padded with spaces to make the output length equal to the input length.

16.3.1.2.15 PDF Payload

This payload extracts PDF payload from the request and lets you define regex to control precise extraction.

The following figure illustrates the PDF payload fields.

Figure 16-33: PDF payload

The properties for the PDF payload are explained in the following table.

Table 16-39: PDF payload

Properties	Description
Pattern	Pattern to be matched for is specified in the field.

Properties	Description
	If no pattern is specified, then the whole input is considered for matching.

Note: The DSG PDF codec supports only text formats in PDFs.

For any assistance in supporting additional text formats, contact Protegility Professional Services.

16.3.1.2.16 Protocol Buffer Payload

The PBpath defines a way to address fields in binary encoded protocolbuf messages. It uses field ids to construct an address messages or fields in a nested message hierarchy.

An example for the PBpath field is shown as follows:

```
1.101.2.201.301.2.401.701.2.802
```

In DSG, protocol buffer version 2 is used.

The following figure illustrates the Protocol Buffer (protobuf) payload fields.



Figure 16-34: Protocol Buffer Payload

The properties for the Protocol Buffer payload are explained in the following table.

Table 16-40: Protocol Buffer Payload

Properties	Description
PBPath List	<p>This PB element value to be extracted is specified in the PB path.</p> <p>Note: Ensure that you enter the PBPath by following proper syntax for extracting the protobuf messages. If you enter incorrect syntax, then the service which has this protobuf payload definition in the rule fails to load and process the request.</p>

16.3.1.2.17 Secure File Transfer Payload

This payload extracts SFTP message from the request and lets you further processing to be done on the files.

The following figure illustrates the Secure File Transfer payload fields.

Payload*	Secured File Transfer
File Name* ⓘ	.*
Method* ⓘ	Download

Figure 16-35: Secured File Transfer Payload

The properties for the Secured File Transfer payload are explained in the following table.

Table 16-41: Secured File Transfer Payload

Properties	Description
File Name	Name of the file to be matched. If the field is left blank, then all the files are matched.
Method	Rule to be applied on the download or the upload of files.

16.3.1.2.18 Amazon S3 Object

This payload extracts Amazon S3 object from the request and lets you define regex to control precise extraction. It is generally used with the Amazon S3 service.

The following figure illustrates the Amazon S3 Object payload fields.

Payload*	Amazon S3 Object
Object Key ⓘ	.*?csv
Target Object* ⓘ	Object Key

Figure 16-36: Amazon S3 Object Payload

The properties for the Amazon S3 Object payload are explained in the following table.

Table 16-42: Amazon S3 Object Payload

Properties	Description
Object Key	Regex logic to identify source object key to be extracted.
Target Object	Object attribute that will be extracted from the following options. <ul style="list-style-type: none"> • Object Key • Object Data

16.3.1.2.19 Shared File

This payload extracts file from the request and lets you define regex to control precise extraction. It is generally used with Mounted services, namely NFS and CIFS.

The following figure illustrates the NFS/CIFS share-related Shared File payload fields.

Payload*	Shared File
File Key ⓘ	.*
Target File* ⓘ	File Data

Figure 16-37: Shared File Payload

The properties for the Shared File payload are explained in the following table.

Table 16-43: Shared File Payload

Properties	Description
File Key	Regex logic to identify source file key to be extracted. Note: Click Test Regex to verify if the regex expression is valid.
Target File	Attribute that will be extracted from the payload. The options are: <ul style="list-style-type: none">• File Key• File Data

16.3.1.2.20 SMTP Message Payload

This payload extracts SMTP payload from the request and lets you define regex to control precise extraction.

The following figure illustrates the SMTP message payload fields.

Name*	COMPANY DATA Etc. > Customer Addition/Search - Customer Company Name Protec
Description	Protects customer name
Enabled	<input checked="" type="checkbox"/>
Action*	Extract
Payload*	SMTP Message
SMTP Message Type* ⓘ	SMTP_MESSAGE
Command ⓘ	
Target Object ⓘ	Command Arguments

Figure 16-38: SMTP Payload

The properties for the SMTP payload are explained in the following table.

Table 16-44: SMTP Payload

Properties	Description
SMTP Message Type	Type of SMTP message to be intercepted.
Method	A condition is applied, if matching is to be performed on the files that are uploaded or the files that are downloaded.
Command	Regular expression to be matched with a command.
Target Object	Attribute to be extracted.

16.3.1.2.21 Text Payload

This payload extracts text payload from the request and lets you define regex to control precise extraction.

The following figure illustrates the Text payload fields.



Payload* Text
Prerequisite Match Patte...
Pattern
Pattern Group Ids +
Group Id* 0
Profile Name
User Comments
Encoding Standard
Codec* utf_8

NOTE: Recursion rule reference detected. Recursions are allowed with the limit of max depth configuration (default to 10)

Figure 16-39: Text Payload

The properties for the Text payload are explained in the following table.

Table 16-45: Text Payload

Properties	Sub-Field	Description
Prerequisite Match Pattern		Regular expression to be matched before the action is executed.
Pattern		The regular expression pattern on which the extraction is applied is specified in this field. For example, consider if the text is “Hello World”, then pattern would be “\w+”.
	Pattern Group Id	The grouping number to extract from the Regular Expression Pattern. For example, for the text “Hello World”, Group Id would be 0 to match characters in first group as per regex.
	Profile Name	Profile to be used to perform transform operations on the matched content.
	User Comments	Additional information related to the action performed by the group processing.
Encoding		Type of encoding to be used.
Codec		The encoding method used is specified in this field. For more information about codec types, refer to the section Standard Encoding Method List .

16.3.1.2.22 URL Payload

This payload extracts URL payload from the request and extract precise object based on selection.

The following figure illustrates the URL payload fields.

Payload* Uniform Resource Locator (URL)
Target Object fragment

Figure 16-40: URL Payload

The properties for the URL payload are explained in the following table.

Table 16-46: URL Payload

Properties	Description
Target Object	Object attribute to be extracted.

16.3.1.2.23 User Defined Extraction Payload

This codec lets you define custom extraction logic and pass arguments to the next rule.

Note:

The language that is currently supported for extraction is Python.

Note:

From DSG 3.0.0.0, the Python version is upgraded to python 3. The UDFs written in Python v2.7 will not be compatible with Python v3.10. To migrate the UDFs from python 2 to python 3, refer to the section [Appendix I: Migrating the UDFs to Python 3](#).

The following figure illustrates the User Defined Extraction payload fields.

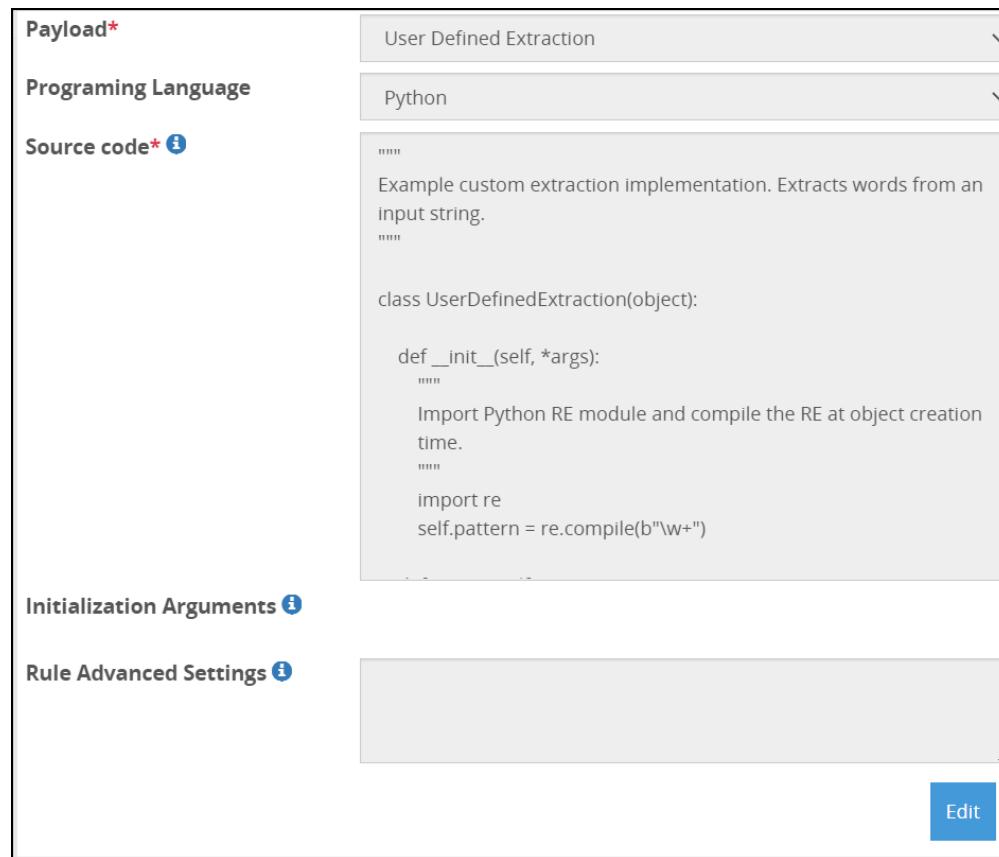


Figure 16-41: User Defined Extraction Payload

The properties for the User Defined Extraction payload are explained in the following table.

Table 16-47: User Defined Extraction Payload

Properties	Description
Programming Language	Programming language used for data extraction is selected. The language that is currently supported for extraction is Python.

Properties	Description
Source Code	<p>Source code for the selected programming language.</p> <p>Caution: Ensure that the class name <i>UserDefinedExtraction</i> is not changed while creating the UDF.</p> <p>Note: For more information about the supported libraries apart from the default Python modules, refer to the section Supported Libraries.</p>
Initialization Arguments	<p>The list of arguments passed to the constructor of the user defined extraction code is specified in this field.</p>
Rule Advanced Settings	<p>As part of the security enhancements, the <i>gateway.json</i> file includes the <i>globalUDFSettings</i> key. This key and the corresponding value defines a list of vulnerable modules and methods that are blocked.</p> <p>Provide the specific module that must be overruled. The module will be overruled only for the extract rule. The parameter must be set to the name of the module that must be overruled in the following format.</p> <pre data-bbox="861 903 1432 967">{ "override_blocked_modules": ["<name of module>" , "<name of module>"] }</pre> <p>Note: Currently, methods cannot be overruled using Advanced settings.</p> <p>For more information about the blocked methods and modules, refer to the section Blocked Modules and Methods in UDF.</p> <p>Using the <i>Rule Advanced Settings</i> option, any module that is set as blocked, can be overruled to be unblocked.</p> <p>For example, setting the value as <code>{"override_blocked_modules": ["os"]}</code> allows the <i>os</i> module to be used in the code in spite of it being blocked in the <i>gateway.json</i> file.</p>

Note: The DSG supports the usage of the PyJwt python library in custom UDF creations. PyJWT is a python library that is used to implement Open Authentication (OAuth) using JSON Web Tokens (JWT). JSON Web Tokens (JWT) is an open standard that defines how to transmit information between a sender and a receiver as a JSON object. To authenticate JWT for OAuth, you must write a custom UDF. The PyJwt library version supported by the DSG is *1.7.1*.

For more information about writing a custom UDF on the DSG, refer to the section [User Defined Functions \(UDF\)](#).

Note: The DSG supports the usage of the Kafka python library in custom UDF creations. Kafka is a python library that is used for storing, processing, and forwarding for applications in a distributed environment. For example, the DSG uses the Kafka library to forward Transaction Metrics logs to external applications. The Kafka library version supported by the DSG is *2.0.2*.

For more information about writing a custom UDF on the DSG, refer to the section [User Defined Functions \(UDF\)](#).

Note: The DSG supports the usage of the Openpyxl Python library in custom UDF creations. Openpyxl is a Python library that is used to parse Excel *.xlsx*, *.xlsm*, *.xltm* files. This library enables column-based transformation for Microsoft Office Excel. The Openpyxl library version supported by the DSG is *2.6.4*.

Note:

The DSG uses the in-built tarfile python module for custom UDF creation. This module is used in the DSG to parse *.tar* and *.tgz* packages. Using the tarfile module, you can extract and decompress *.tar* and *.tgz* packages.

16.3.1.2.24 ZIP Compressed File Payload

This payload extracts ZIP file from the request and lets you extract file name or file content.

The following figure illustrates the ZIP Compressed File payload fields.

Payload*	ZIP Compressed File
File Name ⓘ	
Recurse Zip in Zip	<input checked="" type="checkbox"/>
Target Object ⓘ	File Content

Figure 16-42: ZIP Payload

The properties for the ZIP payload are explained in the following table.

Table 16-48: ZIP Payload

Properties	Description
File Name	Name of the file on which action is to be performed.
Target Object	File name or the file content to be extracted.

16.3.1.3 Log

This section provides an overview of the Log action type.

The Log action type is used to check the contents of the payload in the gateway log for tracking and auditing purposes. To use this functionality, you must create a rule in the child node of the extract rule defined for the payload. The payload can be defined for services, such as, HTTP, SFTP, REST, and so on. When you invoke the extract rule for the payload, the log rule will display the contents of the payload in the gateway log.

The fields for the Logs action type are as seen in the following figure.

Level	Information
Destination	Internal
Title Format i	%(name)s
Message Format i	%(value)s

Figure 16-43: Log Action

The properties for the Log action are explained in the following table.

Table 16-49: Log Action

Properties	Description
Level*	Type of log to be generated. The type of log selected in this field decides the log entry that is displayed on the Audit screen in Forensics.
Destination**	Location where the log file is sent to is specified in this field.
Title Format	Used to format log message title using rule %(name) as a variable. Only applicable to destinations that support title.
Message Format	Used to format log message using %(value) as a variable. Value represent the data extracted by parent rule.

* -In the Log action, depending on the severity of the message, you can define different types of log levels. The following table shows the DSG log level setting and the corresponding log entry that is seen in Forensics.

DSG Log Level	Logging in Forensics
Warning	Normal
Error	High
Verbose	Lowest
Information	Low
Debug	Debug

** The following options are available for the Destination field:

- Forensics: Log is sent to the Forensics and the DSG internal log.
- Internal: Log is sent only to the DSG internal log.

Consider the following example to understand the functionality of the Log action type. In this example, the Log action type functionality is used to find the transactionID from a sample text message in the HTTP payload. The following RuleSet structure is created to find the transactionID:



Figure 16-44: RuleSet structure

- i. Create an extract rule for the HTTP payload using the default RuleSet template defined under the REST API service. The HTTP payload consists of the following sample text message:

```
{
  "source": {
    "merchantId": "LCPLC-lphf21",
    "storeId": "LPHF2100303086",
    "terminalId": "0",
    "operatorId": "10",
    "terminalType": "GASPRO",
    "offline": false
  },
  "message": {
    "transactionType": "WALLET",
    "messageType": "VERIFYPARTNER",
    "transactionTime": "2018-05-09T11:48:47+0000",
    "transactionId": "20180509114847LPHF2100301237000090"
  },
  "identity": {
    "key": "4169671111",
    "entryType": "KEYEDOPERATOR"
  }
}
```

- ii. The fields for the extract rule for the HTTP message payload are as seen in the following figure:

The screenshot shows the configuration interface for an Extract rule. The fields include:

- Name***: Extract HTTP Message
- Description**: (empty)
- Enabled**: checked
- Action***: Extract
- Payload***: HTTP Message
- HTTP Message Type***: Request
- HTTP Method**: POST
- URI**: /logruletest
- Request Headers**: (empty)
- Message Body**: (empty)
- Require Client Certificate**: unchecked
- Authentication**: None

Figure 16-45: Extract rule fields

- iii. Create a child extract rule to extract the text message in the payload.

- a. Create a RegEx expression, "**transactionID":(.*)**", to match the pattern for the transactionID. The fields for the child extract rule are as seen in the following figure:

The screenshot shows the configuration interface for a child Extract rule. The fields include:

- Name***: Extract transactionId from HTTP message body
- Description**: (empty)
- Enabled**: checked
- Action***: Extract
- Payload***: Text
- Prerequisite Match Pattern**: "transactionId".(.*)
- Pattern**: (empty)
- Pattern Group Ids**:
 - Group Id***: 1
 - Profile Name**: (empty)NOTE: Recursion rule reference detected. Recursions are allowed with the limit of max depth configuration (default to 10)
- User Comments**: (empty)

Figure 16-46: Child extract rule fields

- iv. Create another child rule for the Log action type. The fields for the Log action type are as seen in the following figure:

The screenshot shows a configuration interface for a rule named "Log TransactionID Rule". The fields are as follows:

- Name***: Log TransactionID Rule
- Description**: (empty)
- Enabled**: checked
- Action***: Log
- Level**: Warning
- Destination**: Internal
- Title Format**: %(name)s
- Message Format**: %(value)s

Figure 16-47: Log action type fields

- v. Add the following values in the Log action type.
 - a. Action: Log
 - b. Level: Warning
 - c. Destination: Internal
 - d. Title Format: %(name)s
 - e. Message Format: %(value)s
- vi. After the log rule is processed, the following message will be displayed in the DSG internal log:

The screenshot shows a log entry in the DSG internal log. The log entry details are as follows:

Host	PID	Time	Source	Module	Level
2.10.1.5	13357	30-Jan-2019 9:18:54	log	LogInternal	Warning
Message					
transactionId:20180509114847					

Figure 16-48: Message displayed in the DSG internal log

16.3.1.4 Profile Reference

You can refer to an external profile using the Profile Reference action.

Caution:

It is recommended that when creating Rulesets for Dynamic CoP, the *Profile Reference* rule is used for data transformation instead of the *Transform* rule. The security benefits of using *Profile Reference* rule are higher than the *Transform* rule since the requests are triggered on the fly.

The fields for the Profile Reference action are as seen in the following figure.

The screenshot shows a configuration interface for a Profile Reference action. The fields are as follows:

- Action***: Profile Reference
- Profile Name**: Customer Name

Figure 16-49: Profile Reference Action

The following table describes the fields for the Profile Reference action.

Table 16-50: Profile Reference

Field	Description
Profile Name	Select a reference to an external profile from the drop down

Note: The Profile Reference action type must always be created as a leaf node (a rule without any child nodes).

16.3.1.5 Set User Identity

You can set the username using the Set User Identity action.

The fields for the Set User Identity action are as seen in the following figure.



Figure 16-50: Set User Identity Action

Note: If the *Set User Identity* rule is followed by a *Transform* rule, then any data processing logs generated in Forensics are logged with the user set using the *Set User Identity* rule.

In addition, the user set using the *Set User Identity* rule overrides the user set in the *Transform* rules for auditing any data processing logs in Forensics.

Note: If the "Set User Identity" rule is configured along with Basic Authentication, then the "user_name" field in the transaction metrics will be set with the username configured in the "Set User Identity" rule.

If Basic Authentication is configured without the "Set User Identity" rule, then the "user_name" field in the transaction metrics displays the username configured in the Basic Authentication.

16.3.1.6 Set Context Variable

You can use this action type when you want to pass any value as an input to the rule. The value set due to this rule will be maintained throughout the rule lifecycle.

The following table describes the Variable Name type supported by the Set Context Variable option.

Table 16-51: Variable Name Fields

Field	Description
User IP Addr	Captures the client IP address forwarded by the load balancer that distributes client requests among DSG nodes. This IP address is displayed in the Insights Forensics Audit log.
Value-External IV Protect, Unprotect	Uses the External IV value that is sent in the header to protect or unprotect data. This value overrides the value set in the Default External IV field in the Transform rule.
Value-External IV Reprotect	Uses the External IV value that is sent in the header to reprotect data. This value overrides the value set in the Reprotect External IV field in the Transform rule.
Dynamic Rule	Used when <i>Dynamic CoP</i> is implemented for the given Ruleset hierarchy. A request header with Dynamic CoP rule accesses the URI to complete the Ruleset execution.
Client Correlation Handle	Captures the Linux epoch time when the protect or unprotect operation is successful.

Field	Description
User Defined Headers	<p>Extracts JSON data from the input and set it into the response header. The JSON data is extracted into key-value pairs and appended in the response header. This field also accepts list of lists as input. For example, <code>[["access-id", "asds62231231"], ["secret-access-token", "sdas1353412"]]</code>.</p> <p>Consider an example where in some sample JSON data, <code>{"access-id": "asds62231231", "secret-access-token": "sdas1353412"}</code>, is sent from a server to the DSG. After the DSG processes the request, the JSON data is extracted into key-value pairs and appended in the response header. The key will be the header name and the value will be the corresponding header value. The following snippet is displayed in the response header:</p> <pre>access-id -> asds62231231 secret-access-token -> sdas1353412</pre>

Note: The Set Context Variable action type must always be created as a leaf node (a rule without any child nodes).

16.3.1.6.1 User IP Addr

You can record the IP address of the client that sends a request to a DSG node in the Insights Forensics Audit log. When a client request is sent to the load balancer that distributes incoming requests to the cluster of DSG nodes, the load balancer appends a header to the request. This header captures the client IP address.

The types of headers can be *X-Forwarded-For*, which is most commonly used, or *X-Client-IP*, *User-Agent*, and so on.

Before a Set Context Variable with the User IP Addr Variable Name type rule is created, an extract rule that extracts the Header with the given header name, such as X-Forwarded-For, from a request would be created.

If a request header sends an IP address 192.168.0.0 as the X-Forwarded-For value, the following image shows the client IP in the Forensics log displaying this IP address value.

additional_info.description	additional_info.module	additional_info.procedure	additional_info.title	client.ip	client.username	cnt	correlationid	endtime	filetype	index
PCPGAUDIT:				192.168.0.0	alliance	1440		2021-4		
PCPGAUDIT:				10.242.2.3	alliance	9		2021-C		
				10.242.2.3	alliance	1		2021-C		

Figure 16-51: Client IP Address in Forensics Audit log

The fields for the Variable Name type are as seen in the following figure.

Action* Set Context Variable

Variable Name User IP Addr

Truncate Input

Rule Advanced Settings

Figure 16-52: User IP Addr Variable Name type

The following table describes the fields.

Table 16-52: Set Context Variable - User IP Addr

Field	Description	Default (if any)
Truncate Input	Select this check box to truncate any context variable value passed in the header that exceeds the <i>maximumInputLength</i> set in the <i>Rule Advanced Settings</i> . If this check box is not selected and the value set in the context variable exceeds the length set in the <i>maximumInputLength</i> parameter, then the transaction fails with an error.	
Rule Advanced Settings	Set the parameter <i>maximumInputLength</i> such that data beyond this length is not set as the context variable. The datatype for this option is bytes.	512

16.3.1.6.2 Value-External IV Protect, Unprotect

You can send an external IV value that will be used along with the protect or unprotect algorithm in the request header to create more secure encrypted data. External IV values add additional layer of randomness and help in creating secure tokens.

Note: This value overrides the value set in the Default External IV field in the Transform rule.

The fields for the Variable Name type are as seen in the following figure.

The screenshot shows a configuration interface for a rule action. The 'Action*' dropdown is set to 'Set Context Variable'. The 'Variable Name' dropdown is set to 'Value - External IV Protect, Unprotect'. The 'Truncate Input' checkbox is unchecked. The 'Rule Advanced Settings' section is collapsed.

Figure 16-53: Value-External IV Protect, Unprotect

The following table describes the fields.

Table 16-53: Set Context Variable - Value-External IV Protect, Unprotect

Field	Description	Default (if any)
Truncate Input	Select this check box to truncate any context variable value passed in the header that exceeds the <code>maximumInputLength</code> set in the <i>Rule Advanced Settings</i> . If this check box is not selected and the value set in the context variable exceeds the length set in the <code>maximumInputLength</code> parameter, then the transaction fails with an error.	
Rule Advanced Settings	Set the parameter <code>maximumInputLength</code> such that data beyond this length is not set as the context variable. The datatype for this option is bytes.	512

Note: If an External IV value is sent in the header to protect or unprotect sensitive data, with the *case-preserving* and *position-preserving* property enabled in the *Alpha-Numeric (0-9, a-z, A-Z)* token type, then the External IV property is not supported.

For more information about Case-preserving and Position-preserving Tokenization and External IV, refer to the sections *Case-Preserving and Position-Preserving Tokenization* and *External IV* in the [Protection Methods Reference Guide 9.1.0.0](#).

16.3.1.6.3 Value-External IV Reprotect

You can send an external IV value that will be used along with the reprotect algorithm in the request header to create more secure encrypted data. External IV values add additional layer of randomness and help in creating secure tokens.

Note: This value overrides the value set in the Default External IV field in the Transform rule.

The fields for the Variable Name type are as seen in the following figure.

The screenshot shows a configuration interface for a rule action. The 'Action*' dropdown is set to 'Set Context Variable'. The 'Variable Name' dropdown is set to 'Value - External IV Reprotect'. The 'Truncate Input' checkbox is unchecked. The 'Rule Advanced Settings' section is collapsed.

Figure 16-54: Value-External IV Reprotect

The following table describes the fields.

Table 16-54: Set Context Variable - Value-External IV Reprotect

Field	Description	Default (if any)
Truncate Input	Select this check box to truncate any context variable value passed in the header that exceeds the maximumInputLength set in the <i>Rule Advanced Settings</i> . If this check box is not selected and the value set in the context variable exceeds the length set in the maximumInputLength parameter, then the transaction fails with an error.	
Rule Advanced Settings	Set the parameter maximumInputLength such that data beyond this length is not set as the context variable. The datatype for this option is bytes.	512

Note: If an External IV value is sent in the header to protect or unprotect sensitive data, with the *case-preserving* and *position-preserving* property enabled in the *Alpha-Numeric (0-9, a-z, A-Z)* token type, then the External IV property is not supported.

For more information about Case-preserving and Position-preserving Tokenization and External IV, refer to the sections *Case-Preserving and Position-Preserving Tokenization* and *External IV* in the *Protection Methods Reference Guide 9.1.0.0*.

16.3.1.6.4 Dynamic Rule

The Dynamic Rule provides a hook in form of the URI in the preceding rule and a logical endpoint for the Dynamic CoP header request to join the rule tree.

After you define the Dynamic Rule variable name type, you can proceed with creating the *Dynamic Injection* action type.

The fields for the Variable Name type are as seen in the following figure.

The screenshot shows a configuration interface for a 'Dynamic Rule'. The 'Action*' dropdown is set to 'Set Context Variable'. The 'Variable Name' dropdown is set to 'Dynamic Rule'. The 'Truncate Input' checkbox is unchecked. The 'Rule Advanced Settings' section is collapsed.

Figure 16-55: Dynamic Rule

The following table describes the fields.

Table 16-55: Set Context Variable - Dynamic Rule

Field	Description	Default (if any)
Truncate Input	Select this check box to truncate any context variable value passed in the header that exceeds the <i>maximumInputLength</i> set in the <i>Rule Advanced Settings</i> . If this check box is not selected and the value set in the context variable exceeds the length set in the <i>maximumInputLength</i> parameter, then the transaction fails with an error.	
Rule Advanced Settings	Set the parameter <i>maximumInputLength</i> such that data beyond this length is not set as the context variable. The datatype for this option is bytes.	4096

16.3.1.6.5 Client Correlation Handle

The client correlation handle captures the Linux epoch time when the protect or unprotect operation is successful.

When you define rulesets, the rules are structured such that the extract rule identifies the protect successful event from the input message. This rule is followed by the extraction of the timestamp using a UDF rule.

The set context variable rule follows next to set the variable to the extracted timestamp. You can further create a rule that converts this timestamp to a hex value followed by a Log rule to display the exact time of protect and unprotect operation in the ESA Forensics or DSG logs.

The fields for the Variable Name type are as seen in the following figure.

The screenshot shows a configuration interface for a rule. The 'Action*' dropdown is set to 'Set Context Variable'. The 'Variable Name' dropdown is set to 'Client Correlation Handle'. The 'Truncate Input' checkbox is unchecked. Below these fields is a large, empty rectangular area labeled 'Rule Advanced Settings'.

Figure 16-56: Client Correlation Handle

The following table describes the fields.

Table 16-56: Set Context Variable - Client Correlation Handle

Field	Description	Default (if any)
Truncate Input	<p>Select this check box to truncate any context variable value passed to the <i>Set Context Variable</i> rule that exceeds the maximumInputLength parameter value set in the <i>Rule Advanced Settings</i>.</p> <p>Note: The maximum value that can be set for the maximumInputLength parameter value is 20. If this parameter is set to a value greater than 20, then the following warning message appears in the gateway startup logs and the context variable value is truncated to 20 characters.</p> <div style="background-color: #f0f0f0; padding: 5px;"> <p>"Value configured by user is ignored as it exceeds 20 Characters (Maximum Limit)"</p> </div> <p>If this parameter is not configured, the context variable value is truncated to 20 characters by default.</p> <p>If this check box is not selected and the context variable value passed to the <i>Set Context Variable</i> rule exceeds the maximumInputLength parameter value set in the <i>Rule Advanced Settings</i>, then the transaction fails with an error.</p>	
Rule Advanced Settings	<p>Set the parameter maximumInputLength such that data beyond this length is not set as the context variable.</p> <p>The datatype for this option is number of characters.</p>	20

16.3.1.6.6 User Defined Headers

User Defined Headers are meant to provide additional information about an HTTP Response Header that can be helpful for troubleshooting purposes. The User Defined Headers can include information such as custom cookies, state information, and provide information to the load balancer, for example, CPU utilization of a particular node behind the load balancer.

The fields for the Variable Name type are as seen in the following figure.

Figure 16-57: User Defined Headers

The following table describes the fields.

Table 16-57: Set Context Variable - User Defined Headers

Field	Description	Default (if any)
Truncate Input	Select this check box to truncate any context variable value passed in the header that exceeds the <code>maximumInputLength</code> set in the <i>Rule Advanced Settings</i> . If this check box is not selected and the value set in the context variable exceeds the length set in the <code>maximumInputLength</code> parameter, then the transaction fails with an error.	
Rule Advanced Settings	Set the parameter <code>maximumInputLength</code> such that data beyond this length is not set as the context variable. The datatype for this option is bytes.	4096

16.3.1.7 Dynamic Injection

After you define a Dynamic Rule variable name type to store the request header that sends the Dynamic CoP structure, you must configure the Dynamic Injection action type to process Dynamic rules and protect data on-the-fly.

Caution:

It is recommended that when creating Rulesets for Dynamic CoP, the *Profile Reference* rule is used for data transformation instead of the *Transform* rule. The security benefits of using *Profile Reference* rule are higher than the *Transform* rule since the requests can be triggered out of the secure network perimeter of an organization.

For more information about Dynamic CoP, refer to the section [Dynamic Configuration over Programming \(CoP\)](#).

The fields for the Dynamic Injection action are as seen in the following figure.

Action* Dynamic Injection

Authorized Rule Types ⓘ

Rule Type ⓘ	Extract
Payload ⓘ	.*

Rule Type ⓘ

Method ⓘ	Transform
	.*

Figure 16-58: Dynamic Injection Action

The following table describes the Authorized Rule Types applicable to the Dynamic Injection action.

Table 16-58: Dynamic Injection Fields

Field	Description
Rule Type	<p>Allowed Action type that will be parsed from the Dynamic Rule request header.</p> <p>The following are the Available Action type options:</p> <ul style="list-style-type: none"> • Error • Exit • Extract • Log • Profile Reference • Set Context Variable • Set User Identity • Transform
Payload	<p>Select the allowed Payload type that will be parsed from the Dynamic Rule request header. This field also accepts Regex patterns for selecting payload that needs to be parsed.</p> <p>Note: To select all payloads, click the Payload drop down and select ALL.</p> <p>For more information about available Payload options for each Action type, refer to the section Rules and Action types.</p>

Note: The Dynamic Injection action type must always be created as a leaf node (a rule without any child nodes).

16.3.1.7.1 Creating a Dynamic Injection Rule

The dynamic injection rule must be created to send dynamic Ruleset requests.

► To create a Dynamic Injection Rule:

1. Create an extract rule under an existing profile or a new profile to extract the request header with the Dynamic Rule.

Active

Name*: extractheader

Descript...

Enabl...:

Action*: Extract

Payload*: HTTP Message

HTTP Message Type* i: Request

HTTP Method i

URI i: /dynamic/injection

Request Headers i

Message Body i

Require Client Certifica... i

Authentication i: None

Target Object i: Message Header

Header Name i: X-Dynamic-Rule

Header Value i

Target Object i: Value

Figure 16-59: Dynamic Injection Rule-1

- Under this extract rule, create a Set Context Variable rule to extract the Dynamic rule and set as a variable.

Active

Name*: applyrule

Descript...

Enabl...:

Action*: Set Context Variable

Variable Name i: Dynamic Rule

Figure 16-60: Dynamic Injection Rule-2

- Under the same profile, create an extract rule to extract the request message.

The screenshot shows the configuration of a Dynamic Injection rule. The rule is named "dynamicinjection". It is set to be active and enabled. The action is "Extract" from an "HTTP Message" (Request) with the URI "/dynamic/injection". The target object is the "Message Body".

Name*	dynamicinjection
Descript...	
Enabl...	<input checked="" type="checkbox"/>
Action*	Extract
Payload*	HTTP Message
HTTP Message Type* ⓘ	Request
HTTP Method ⓘ	
URI ⓘ	/dynamic/injection
Request Headers ⓘ	
Message Body ⓘ	
Require Client Certifica... ⓘ	
Authentication ⓘ	None
Target Object ⓘ	Message Body

Figure 16-61: Dynamic Injection Rule-3

- Under the extract rule created in step 3, create the Dynamic Injection rule. The Dynamic rule received in the request header in step 2 will be hooked to this rule for further processing.

The screenshot shows the configuration of a Dynamic Injection rule. The rule is named "ApplyInjection". It is set to be active and enabled. The action is "Dynamic Injection". The rule type is "Extract" with payload "*". The method is "Transform" with payload "*".

Name*	ApplyInjection
Descript...	
Enabl...	<input checked="" type="checkbox"/>
Action*	Dynamic Injection
Authorized Rule Types ⓘ	
Rule Type ⓘ	Extract
Payload ⓘ	*
Rule Type ⓘ	Transform
Method ⓘ	*

Figure 16-62: Dynamic Injection Rule-4

- Click **Deploy** or **Deploy to Node Groups** to apply configuration changes.
When a request is received at the configured URI, the request header and request body are processed, and then the response is sent.

16.3.1.8 Error

In HTTP protocol, you can send custom response messages for requests with invalid content, while for other protocols, such as SFTP or SMTP the connection is terminated.

The fields for the Error action are as seen in the following figure.

The screenshot shows a configuration interface for an 'Error' action. On the left, there is a label 'Action*' and a dropdown menu set to 'Error'. Below it is a label 'Message' with a blue information icon, and a text input field containing the placeholder text 'This is error message %%'.

Figure 16-63: Error Action

The following table describes the fields applicable to the Error action.

Table 16-59: Error Action Fields

Field	Description
Message	<p>Add a custom response message for any invalid content.</p> <p>You can add a custom response message for any invalid content using one of the following options.</p> <ul style="list-style-type: none"> • Input an HTTP response message formatted string into this action type by yielding it from the parent rule. <ul style="list-style-type: none"> • Example - 'Error message yielded by parent - %(input)s'. The %(input)s value in the HTTP error response message body carries the string yielded by the parent extraction rule. • Embed the message in the text box. The string must be formatted as an HTTP RFC compliant response message with status-line, header-fields, and body. <ul style="list-style-type: none"> • Example - 'HTTP/1.1 999 EXAMPLE-ERR MyHeader1: Time MyHeader2: 1116 This is an error response body'

Note: The Error action type must always be created as a leaf node (a rule without any child nodes).

16.3.1.9 Transform

The transformation of data depends on the method type applied on the data.

For any Transform action, if you click the () icon to add a new rule, a message TRANSFORM rule cannot have a child rule under it appears.

The fields for the Transform action are as seen in the following figure.

Name*	Transform Rule
Description	
Enabled	<input checked="" type="checkbox"/>
Action*	Transform
Method*	Protegility Data Protection
Protection Method*	Protect
Data Element Name* <small>(i)</small>	
Encryption Data Element <small>(i)</small>	<input type="checkbox"/>

Figure 16-64: Transform Action

The following table describes the methods applicable to the Transform action.

Table 16-60: Transform Action Fields

Field	Description
Protegility Data Protection	Protect data using Protegility Data Protection methods such as tokenization or encryption.
Regular Expression Replace	List the patterns to replace for regular expression transformation.
User Defined Transformation	Provide User Defined Functions (UDFs) for transformation.
GNU Privacy Guard (GPG)	Enable GPG encryption and decryption for data transformation.
SAML Codec	Enable Security Assertion Markup Language (SAML) support.

Note: The Transform action type must always be created as a leaf node (a rule without any child nodes).

Note: If an External IV value is configured in the Transform rule, with the *case-preserving* and *position-preserving* property enabled in *Alpha-Numeric (0-9, a-z, A-Z)* token type, then the External IV property cannot be used to transform sensitive data.

For more information about Case-Preserving and Position-Preserving Tokenization and External IV, refer to the sections *Case-Preserving and Position-Preserving Tokenization* and *External IV* in the [Protection Methods Reference Guide 9.1.0.0](#).

16.3.1.9.1 Protegility Data Protection Method

The fields related to the Protegility Data protection transform rule are listed in this section.

The following table describes the fields for Protegility Data Protection method.

Table 16-61: Protegility Data Protection Method

Field	Sub-Field	Description
Protection Method		Specify the action performed (protection, unprotection, or re-protection).
Data Element Name		Specify the Data element used (for protection, unprotection, or re-protection). <p>Note: The Data Element Name drop down list populates data elements from the deployed policy.</p>
Encryption Data Element		Select to process the encryption data element

Field	Sub-Field	Description
Default External IV		Default value to be used as an external initialization vector.
Reprotect New Data Element Name		<p>New data element name that will be used to reprotect data</p> <p>Note: The Data Element Name drop down list populates data elements from the deployed policy.</p>
Reprotect New Default External IV		New default value to be used as an external initialization vector.
Default Username		<p>Policy Username used for the user.</p> <p>Note: Ensure that you do not use the OS user <i>alliance</i> as a policy user.</p> <p>Note: If the user is not specified, by default the data security operation will be performed by the <i>alliance</i> user.</p>
Encoding		Encoding method to be used.
Codec		<p>Based on the encoding selected, select the codec to be used.</p> <p>For more information about codec types, refer to the section Codecs.</p>
Prefix		Prefix text to be padded before the protected value. This helps in identifying protected text from clear text.
Suffix		Suffix text to be padded after the protected value. This helps in identifying protected text from clear text.
Padding Character		Characters to be added to raise the number of characters to the minimum required size by the Protection method.
Minimum Input length		<p>Number of characters that define if input is too short for the Protection method to be padded with the Padding Character.</p> <p>Note: If a data element is created with the <i>Length Preservation</i> property and it allows short data, then the input data will be tokenized if the input data is short.</p> <p>Note: If a data element is created with the <i>Length Preservation</i> property and it does not allow short data, then the input data will not be tokenized and an error message will be generated.</p>

Field	Sub-Field	Description
Advanced Settings	Permissive Error Handling	<p>Click  to expand.</p> <p>The underlying Application Protector API used in the DSG might encounter input that is not ideal in nature, such as, input too short. In such cases, you can use this option to decide how such inputs should be handled gracefully, as opposed to failing the entire data body processing.</p>
	Enabled	Select to enable permissive handling of error generated due to distorted input.
	Error strings	<p>Regex pattern to identify the errors that need to be handled permissively. You can also provide the exact error message.</p> <p>For example, if the error message on the <i>Log viewer</i> screen is "The input is too short", then you can enter the exact message "The input is too short" in this field. Other error message examples are "The input is too long", "License is invalid", "Permission denied", "Policy not available", and so on.</p> <p>Based on the error message that you encounter and want to handle differently, the value in this field should be adjusted accordingly.</p> <p>For example, a pattern, such as, too short, too long, Permission denied can be used to gracefully handle the respective three errors.</p>
	Output Data	<p>Regex substitution pattern that dictates how output values for erroneous input values are substituted.</p> <p>For example, if this value is set to "?????", then the distorted input will be replaced with this value, thus allowing the rule to process instead of failing due to distorted input. Users may choose such fixed substitution strings to spot individual erroneous input data values post processing of data.</p> <p>You can also add prefix and suffix to the input. The regex must follow the "<code><prefix>\g<0><suffix></code>" REGEX substitution pattern.</p> <p>For example, if you want the input to be identified with the "#*_ as the input prefix and *_#" as the input suffix, the regex pattern with be "#*_\g(0)_*#".</p>

16.3.1.9.2 Regular Expression Replace

The regular expression transform rule related options are listed in this section.

The following table describes the fields for Regular Expression Replace method.

Table 16-62: Regular Expression fields

Field	Sub-Field	Description
Replace Pattern		List of patterns to be matched and replaced for regular expression transformation.
	Match Pattern	Regex logic that defines pattern to be matched. Note: Instead of using .*, use .+ to match the sequence of characters.
	Replace Value	Value to replace the matched pattern.

16.3.1.9.3 User Defined Transformation

The UDF transform rule-related options are listed in this section.

Note:

The language that is currently supported for transformation is Python.

Note:

In DSG 3.0.0.0, the Python version is upgraded to python 3. The UDFs written in Python v2.7 will not be compatible with Python v3.10. To migrate the UDFs from python 2 to python 3, refer to the section [Appendix I: Migrating the UDFs to Python 3](#).

The following figure illustrates the User Defined Transformation payload fields.

Action*	Transform
Method*	User Defined Transformation
Programming Language	Python
Source Code* ⓘ	<pre>class UserDefinedTransformation(object): def transform(self, context): output = [] for c in context["input"].decode(): if c.islower(): output.append(c.upper()) else: output.append(c.lower()) context["output"] = "".join(output).encode()</pre>
Initialization Arguments ⓘ	
Rule Advanced Settings ⓘ	

Figure 16-65: User Defined Transformation Payload

The following table describes the fields for User Defined Transformation method.

Table 16-63: User Defined Transformation Payload

Properties	Description
Programming Language	Programming language used for data transformation is selected. The language that is currently supported for transformation is Python.
Source Code	<p>Source code for the selected programming language.</p> <p>Caution: Ensure that the class name <i>UserDefinedTrasnformation</i> is not changed while creating the UDF.</p> <p>Note: For more information about the supported libraries apart from the default Python modules, refer to the section Supported Libraries.</p>
Initialization Arguments	The list of arguments passed to the constructor of the user defined transformation code is specified in this field.
Rule Advanced Settings	As part of the security enhancements, the <i>gateway.json</i> file includes the <i>globalUDFSettings</i> key. This key and the corresponding value defines a list of vulnerable modules and methods that are blocked.

Properties	Description
	<p>Provide the specific module that must be overruled. The module will be overruled only for the extract rule. The parameter must be set to the name of the module that must be overruled in the following format.</p> <pre data-bbox="861 270 1428 327">{ "override_blocked_modules": ["<name of module>", "<name of module>"] }</pre> <p>Note: Currently, methods cannot be overruled using Advanced settings. For more information about the blocked methods and modules, refer to the section Blocked Modules and Methods in UDF.</p> <p>Using the <i>Rule Advanced Settings</i> option, any module that is set as blocked can be overruled to be unblocked. For example, setting the value as <code>{"override_blocked_modules": ["os"]}</code> allows the <code>os</code> module to be used in the code in spite of it being blocked in the <code>gateway.json</code> file.</p>

Note: The DSG supports the usage of the PyJwt python library in custom UDF creations. PyJWT is a python library that is used to implement Open Authentication (OAuth) using JSON Web Tokens (JWT). JSON Web Tokens (JWT) is an open standard that defines how to transmit information between a sender and a receiver as a JSON object. To authenticate JWT for OAuth, you must write a custom UDF. The PyJwt library version supported by the DSG is *1.7.1*.

For more information about writing a custom UDF on the DSG, refer to the section [User Defined Functions \(UDF\)](#).

Note: The DSG supports the usage of the Kafka python library in custom UDF creations. Kafka is a python library that is used for storing, processing, and forwarding for applications in a distributed environment. For example, the DSG uses the Kafka library to forward Transaction Metrics logs to external applications. The Kafka library version supported by the DSG is *2.0.2*.

For more information about writing a custom UDF on the DSG, refer to the section [User Defined Functions \(UDF\)](#).

Note: The DSG supports the usage of the Openpyxl Python library in custom UDF creations. Openpyxl is a Python library that is used to parse Excel `.xlsx`, `.xlsm`, `.xltm`, `.xltx` files. This library enables column-based transformation for Microsoft Office Excel. The Openpyxl library version supported by the DSG is *2.6.4*.

16.3.1.9.4 GNU Privacy Guard (GPG)

GPG software is used to encrypt and decrypt data using a public and private key. The GPG encrypted data is first optionally compressed, encrypted with a one-time generated session key, and this session key is then encrypted with the public key. The extracted data from execution of RuleSet can be transformed using the GPG method in the Transform action.

From the DSG Web UI, in the **Operation** field, you can either select Encrypt or Decrypt operation. The options for each operation vary based on the selection.

Note: The DSG appliance is compatible with GPG v2.1. For more information about GPG v2.1, refer to the GPG documentation at <https://www.gnupg.org/faq/whats-new-in-2.1.html>

16.3.1.9.4.1 Importing keys

If you are using public and private key generated outside of the DSG, you can use the following steps to import these keys into the DSG.

► To import keys:

1. Upload the public and private key in the DSG Web UI by navigating to **Cloud Gateway > Transport > Certificate/Key Material**.

The *Certificate/Key Material* screen appears.

2. On the *Certificate/Key Material* screen, click **Upload**.

3. Click **Choose File** and select the public key to be uploaded.

4. Similarly, click **Choose File** and select the private key to be uploaded.

5. On the DSG CLI Manager, navigate to the `/opt/protegility/alliance/config/resources` directory.

6. Import the public key by running the following command:

```
gpg --homedir /opt/protegility/alliance/config/resources --import <public_key_file_name>
```

7. Import the private key by running the following command:

```
gpg --homedir /opt/protegility/alliance/config/resources --allow-secret-key-import --pinentry-mode loopback --import <private_key_file_name>
```

8. Trust the imported keys as ultimate keys by running the following command:

```
gpg --homedir /opt/protegility/alliance/config/resources --edit-key {KEY_ID}
gpg> trust
#enter 5<RETURN>
#enter y<RETURN>
gpg> quit
```

16.3.1.9.4.2 Generating GPG keys

This section explains the steps to generate GPG keys.

► To generate GPG keys on ESA:

1. Login to ESA CLI, and then navigate to **Administration > OS Console**.

2. Navigate to the `/opt/protegility/alliance/config` directory.

3. Execute the following command to start generating a key.

```
gpg --homedir /opt/protegility/alliance/config/resources/ --pinentry-mode=loopback --full-generate-key
```

4. Select the type of key that you want to generate from the following options.

- (1) RSA and RSA (default)
- (2) DSA and Elgamal
- (3) DSA (sign only)
- (4) RSA (sign only)

5. Enter the keysize for the key. The keysize can range between 1024 to 4096.

6. Select the validity of the key from the following options.

- 0 = key does not expire
- <n> = key expires in n days
- <n>w = key expires in n weeks
- <n>m = key expires in n months
- <n>y = key expires in n years

7. Enter the real name that identifies the key.

8. Enter the email address for the key.

9. Enter a comment for the key.

Note: The public key in GPG includes a key and user ID information that identifies the key with the user ID.

10. Select (*O*) to confirm the user ID details.

11. Press Enter or provide a passphrase. The passphrase is used during decryption.

12. Run the following command to verify the key is generated.

```
gpg --homedir /opt/protegility/alliance/config/resources --list-keys
```

The gpg directory must include the following files after you generate a GPG key successfully:

- pubring.gpg
- secring.gpg
- trustdb.gpg
- random_seed
- s.gpg-agent
- s.gpg-agent.ssh
- s.gpg-agent.extra
- s.gpg-agent.browser
- private-keys-v1.d
- openpgp-revocs.d

16.3.1.9.4.3 Encrypt operation

The encrypt operation transform rule related options for GPG rule implementation are listed in this section.

The following table describes the fields for Encrypt operation in the GNU Privacy Guard method.

Table 16-64: Encrypt Operation fields

Field	Description	Restrictions (if any)
Recipient Name	Encrypt data for the user provided. Recipient name is defined when the public key is generated. You can either provide the email id or the key id.	
ASCII Armor*	Enable to generate ASCII format output. This option can be used when the output data needs to be created in a format that can be safely sent via email or be printed.	
Custom Arguments	Provide additional arguments that you want to pass to the GPG command line apart from the given arguments. Ensure that the syntax is correct.	Provide additional arguments that you want to pass to the GPG command line apart from the given arguments. Ensure that the syntax is correct.



Field	Description	Restrictions (if any)
		<ul style="list-style-type: none"> • "--armor" • "--recipient" • "--decrypt" • "--homedir" • "--passphrase" • "--batch" • "--no-tty" • "--no-auto-check-trustdb" • "--no-permission-warning" • "--encrypt" • "--pinentry-mode loopback"

16.3.1.9.4.4 Decrypt operation

The decrypt operation transform rule-related options for the GNU Privacy Guard (GPG) rule implementation are listed in this section.

The following table describes the fields for the *Decrypt* operation in the GPG method.

Table 16-65: Decrypt Operation fields

Field	Description
Passphrase	<p>Provide the private key passphrase as a string or name of the file placed in /config/resources directory that contains the passphrase. A null value means that the private key is not passphrase protected.</p> <div style="background-color: #e0f2e0; padding: 10px; margin-top: 10px;"> <p>Note:</p> <p>When you click the (view)  icon, an encrypted password is displayed.</p> </div> <div style="background-color: #ffd700; padding: 10px; margin-top: 10px;"> <p>Caution:</p> <p>If an older ruleset configuration .zip created using any older DSG version, that includes a GPG ruleset with key passphrase defined, is imported, then the DSG does not encrypt the key passphrase.</p> </div>
Delimiter	Regular Expression used to delimit stream. Rules will be invoked on delimited streams.
Custom Arguments	Provide additional arguments that you want to pass to the GPG command line apart from the given arguments. Ensure that the syntax is accurate.

16.3.1.9.5 Security Assertion Markup Language (SAML) Codec

With support for SAML, you can manage user authentication and authorization uniformly for multiple applications on the same network. Any SAML implementation involves the following two entities, namely, the Single Sign-On (SSO) application or the Service Provider that the user is trying to access and the Identity Provider (IDP) responsible for authenticating the user.

A typical SAML implementation is illustrated in the following diagram.

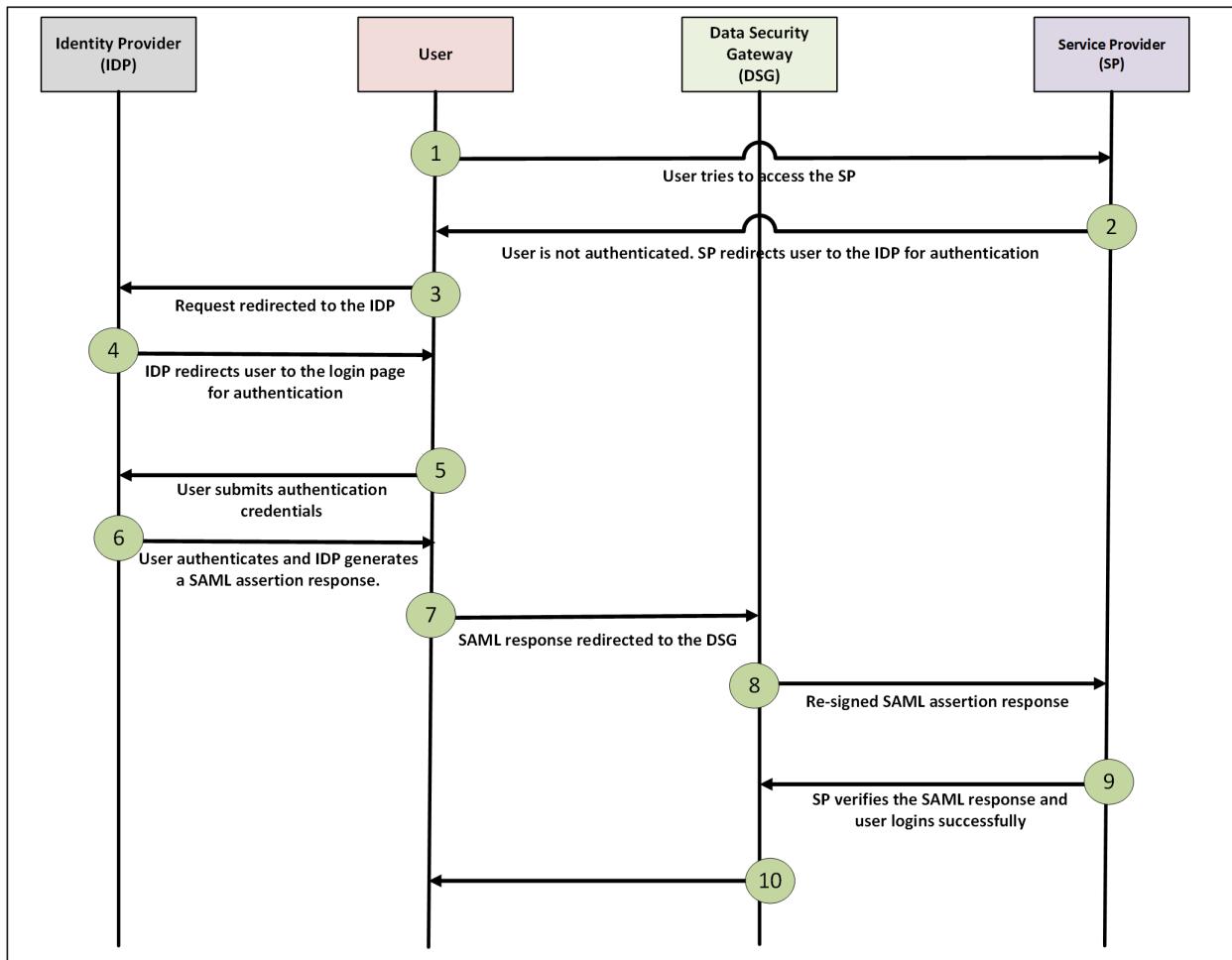


Figure 16-66: SAML SP-DSG-IDP flow

Consider that the user wants to access the Service Provider (SP). The SP redirects the user's authentication request to the Identity Provider (IDP) through the DSG.

When a user tries to access a Service Provider (SP), an authentication request is generated. The DSG receives and forwards the request to the IDP. The IDP authenticates the user and reverts with a SAML assertion response. The DSG updates the domain names in the SAML assertion response, verifies the inbound SAML signature, and re-signs the modified SAML response to ensure that the response is secure and not tampered. The response is then forwarded to the SP. The SP verifies the response and authenticates the user.

After this authentication is complete, when the same user tries to access any other SP on the same network, authentication is no longer required. The IDP identifies the user from the previous active session and reverts with a SAML response that user is already authenticated.

Note:

SAML codec has been tested with the following SP's and IDP's:

- **IDP** – Microsoft Azure AD
- **SP** - SalesForce

Before you begin:

Ensure that the following prerequisites are met:

- The IDP's public certificate, which was used to sign the message response, is uploaded to the ESA using the *Certificate* screen.
- The certificate and private key that the customer wants to use to re-sign the SAML response using the DSG must be uploaded to the ESA.
- The certificate that is used to re-sign the message must be uploaded to the SP for validating the response.
- As the user requests to access the SP are redirected to the IDP through the DSG, ensure that in the SP configurations, the IDP redirect URLs are configured.

For example, if the IDP is <https://login.abc.com>, then in the SP configurations ensure that the redirect URLs are set to <https://secure.login.abc.com>.

- As the SAML response from the IDP, that is used to authenticate the user, is redirected through the DSG to the SP, ensure that the IDP configurations are set as required.

For example, if the SP is <https://biloxi.com>, then in the IDP configurations, ensure that the redirect URLs are set to <https://secure.biloxi.com>.

Note:

After you upload the certificates to the ESA, ensure that you deploy configurations from the ESA such that the certificates and keys are pushed to all the DSG nodes in the cluster.

The SAML codec fields can be seen in the following image.

Action*	Transform
Method*	SAML Codec
Message Verification Certificate* ⓘ	admin
Message Signing Certificate* ⓘ	admin
Assertion Namespace ⓘ	
Key Passphrase ⓘ	
Replace Regex ⓘ	
Replace value ⓘ	
Assertion Settings ⓘ <input checked="" type="checkbox"/>	
Use Message Certificates ⓘ <input checked="" type="checkbox"/>	
Verification Certificate ⓘ	admin
Signing Certificate ⓘ	admin

Figure 16-67: SAML Codec

The following table describes the fields for the SAML codec.

Table 16-66: SAML Codec fields

Field	Sub-Field	Description
Message Verification Certificate		<p>Select the IDP's public certificate from the list of available certificates drop down.</p> <p>Note: Ensure that the certificates are uploaded to the ESA from the <i>Certificate</i> screen.</p>

Field	Sub-Field	Description
Message Signing Certificate		<p>Select the certificate that will be used to re-sign the response from the list of available certificates drop down list. Both message and assertion signing is supported.</p> <p>Note: Ensure that the certificates are uploaded to the ESA from the <i>Certificate</i> screen.</p>
Assertion Namespace		<p>Assertion namespace value defined in the SAML response.</p> <p>Note: This field is redundant. Any value that you enter in this field, will be bypassed.</p>
Key Passphrase		<p>The passphrase for the encrypted signing key that will be used to re-sign the certificate.</p> <p>Note: This field is redundant. Any value that you enter in this field, will be bypassed.</p>
Replace Regex		Regex pattern to identify the forwarded hostname received from the IDP. You can also provide the exact hostname.
Replace value		Hostname that will be used to forward the SAML response.
Assertion settings	Use Message Certificates	<p>Select this checkbox to use the same certificates that were used for verification of the SAML response to re-sign the assertions.</p> <p>Note: This field is redundant. Any value that you enter in this field, will be bypassed.</p>
	Verification Certificate	<p>If you choose to use a certificate other than the one used to re-sign the message response, then select a certificate from the list of available certificates drop down list.</p> <p>Note: This field is redundant. Any value that you enter in this field, will be bypassed.</p>
	Signing Certificate	<p>If you choose to use a certificate other than the one used to re-sign the message response, then select a certificate from the list of available certificates drop down list.</p> <p>Note:</p>

Field	Sub-Field	Description
		This field is redundant. Any value that you enter in this field, will be bypassed.

Chapter 17

DSG REST API

[17.1 Overview](#)

[17.2 REST API Authentication](#)

[17.3 Protecting an XML Document through DSG REST API](#)

[17.4 Java Client using Native java.net.HttpURLConnection](#)

[17.5 Python Client](#)

[17.6 Scala Client using Apache HTTP Client](#)

[17.7 Postman \(Chrome Plugin\) Client](#)

This section provides an overview of Data Security Gateway (DSG) REST API functionality. In addition to providing a conceptual overview, the discussion steps through a use case where DSG is configured with a RuleSet object behind a REST API URL end-point.

The RuleSet has been designed to provide fine-grained protection of an XML document where the sensitive data is identified through an XPath. The sections also contain code samples in Java, Python and Scala that show example invocation of the API from client applications written in these languages.

17.1 Overview

In addition to offering In-Band mechanism for data security, DSG offers a RESTful Web Service API. This mechanism of utilizing DSG's capabilities is referred to as On-Demand data security.

Client applications invoke DSG's REST API by sending it HTTP requests pointed at pre-configured URLs in the DSG. These URLs are internally backed by request processing behaviors which are specified through RuleSets configuration objects. In general, RuleSets allow a user to define hierarchical, step-by-step processing of data conveyed in any Layer-7 protocol messages. Example Rule nodes of a RuleSet tree might include extraction of HTTP POST request body, parsing of the body content according to a certain format (typically specified in HTTP Content-Type header), extraction of sensitive data within the message body and protection of extracted data.

While invoking a REST API call on DSG, a client conveys a document of certain format embedded in an HTTP request method (e.g. HTTP POST) to a pre-configured DSG REST URL. As an outcome of processing the request, DSG sends a 200 response to the client. The response message carries a modified version of the original document wherein select sensitive data pieces in plaintext are substituted with their cryptographic equivalent (either cipher text or tokens).

The following figure shows an example usage of DSG's REST API. It illustrates a fine-grained data security scenario where DSG accepts an incoming request from a client where the request carries a document with certain sensitive data elements embedded in it. DSG parses the input document, extracts sensitive data pieces in it and protects those extracted data fragments as per preconfigured data security policies.

The protected data fragments are substituted in-place at the same location of their plaintext equivalents in the original decoded document. The decoded document is then encoded back to its original format and delivered back to the client as part of the API response.

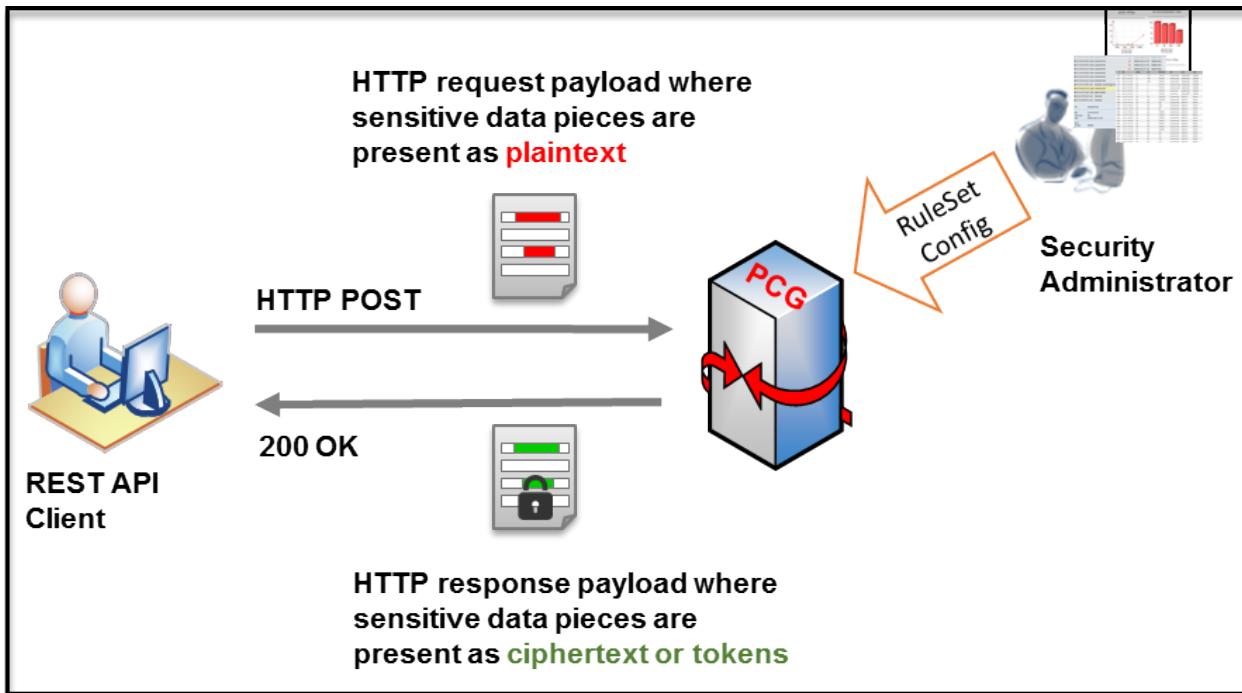


Figure 17-1: Cloud Gateway as RESTful server

While the illustration above shows data protection, one can certainly conclude that the same mechanism can be used for any other form of data transformation natively available in Protegility core data security platform including data un-protection, re-protection, masking and hashing.

17.2 REST API Authentication

DSG supports basic authentication and mutual authentication.

17.2.1 Basic Authentication

DSG supports user authentication using the HTTP basic authentication mechanism.

The user credentials (username and password) are validated against ESA LDAP, which may be connected to any external directory source. Successful user authentication results are cached for a configurable period thus saving the authentication round trips for performance efficiency reasons. The identity of the authenticated users is automatically used as their 'policy user' identity for performing data security operations.

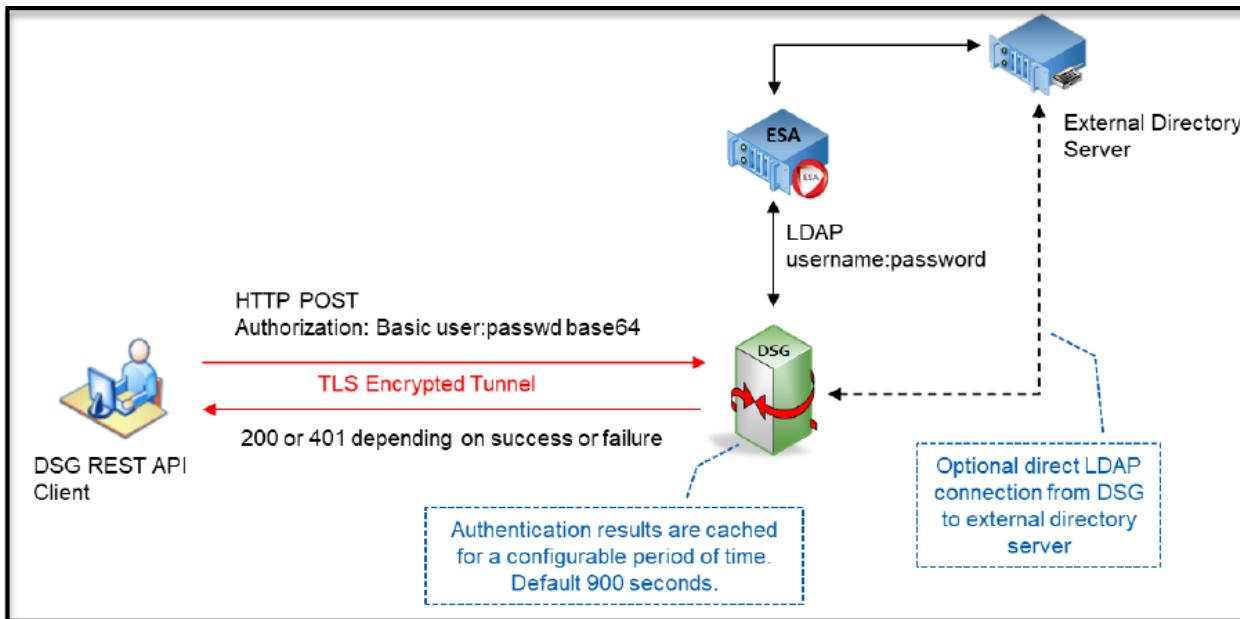


Figure 17-2: REST API Basic Authentication

In this authentication type, the username and password are included in the authorization header.

For permissions and access control, there are three types of roles in DSG:

- Cloud Gateway Admin
- Cloud Gateway Viewer
- Cloud Gateway Auth

The following table describes each of these roles.

Table 17-1: Different types of roles in DSG

Role	Description
Cloud Gateway Admin	Role to allow to modify configurations
Cloud Gateway Viewer	Role to allow to only read the configurations
Cloud Gateway Auth	Role to define user authentication for REST and HTTP protocols

In DSG, you can allow users linked to the Cloud Gateway Authentication role to perform REST API authentication. For a REST API call, the client sends the username and password in the authorization header to the server. After the authentication is successful, all the rules that are part of the REST API call are processed. The authentication is performed at the parent level of every rule and the authentication is cached for every child rule.

The following steps explain the process for the REST API authentication:

1. The user makes a REST API call to the gateway.
2. The authorization parameters, username and password, are verified against LDAP.
3. On successful authentication, the username-password combination is cached for the transaction.
4. The gateway then sends a response to the REST API call sent by the user.
5. If the authentication fails, the server sends a 401 error response stating that the authorization parameters are invalid. The REST API call then terminates.

To enable REST API authentication from the ESA Web UI, ensure that the user is linked to the Cloud Gateway Auth role as shown in the following figure.

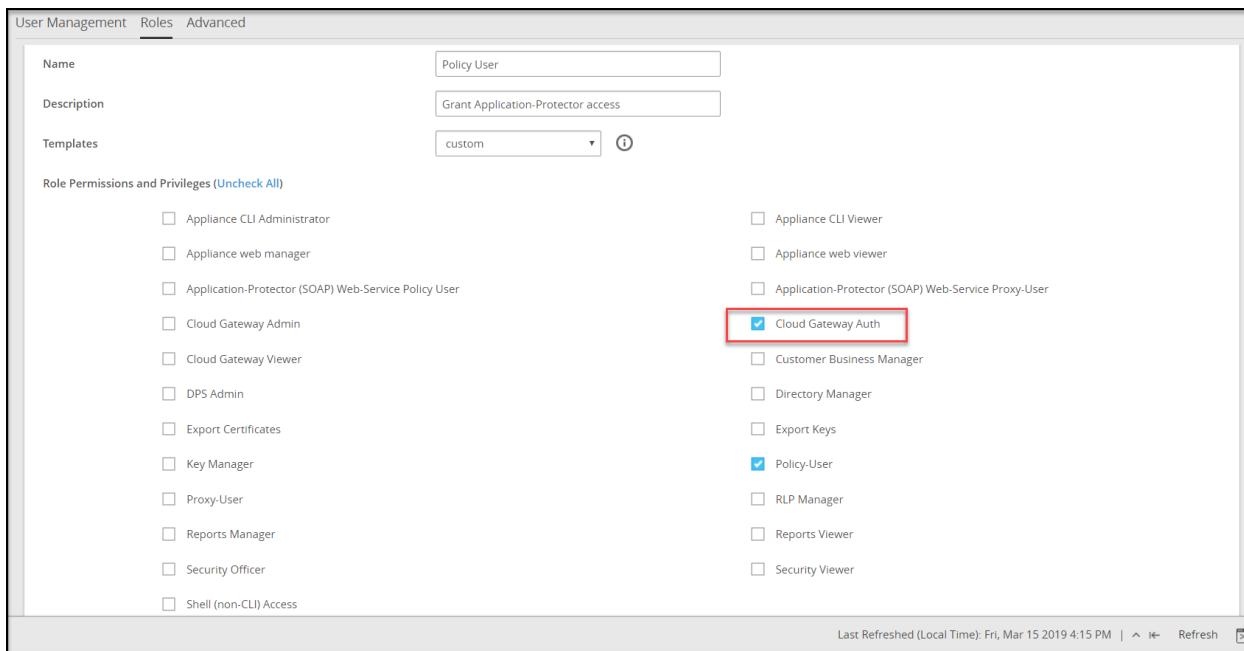


Figure 17-3: REST API Authentication

17.2.1.1 Enabling Rest API Authentication

You can enable REST authentication using the HTTP message payload.

► To enable REST API Authentication:

1. In the ESA Web UI, navigate to **Cloud Gateway > RuleSet**.
2. Select the required rule.
3. In the **Authorization** drop down list, select **Basic**.

The screenshot shows the configuration interface for a rule named "Text Protection 1". The "Target Object" dropdown at the bottom is set to "Basic", which is highlighted in blue.

Setting	Value
Name*	Text Protection 1
Description	(empty)
Enabled	<input checked="" type="checkbox"/>
Action*	Extract
Payload*	HTTP Message
HTTP Message Type*	Request
HTTP Method	POST
URI	/protect/text\$
Request Headers	(empty)
Message Body	(empty)
Require Client Certifica...	<input type="checkbox"/>
Authentication	None
Target Object	Basic

Figure 17-4: Authorization Settings

- Save the changes.

You can also request client certificates as a part of authorization to make a REST API call. After the certificate is verified successfully, all the rules that are a part of the REST API call are executed. If the certificate is invalid, a 401 error response is sent to the user.

17.2.1.2 Enabling a Client Certificate for a Rule

Apart from specifying client certificate at tunnel level, you can also specify client certificate at service level.

► To enable Client Certificate for a Rule

- In the ESA Web UI, navigate to **Cloud Gateway > RuleSet**
- Select the required rule.
- Check the **Require Client Certificate** checkbox.

The screenshot shows the configuration of a 'Text Protection 1' rule. Key settings include:

- Name***: Text Protection 1
- Description**: (empty)
- Enabled**: Checked
- Action***: Extract
- Payload***: HTTP Message
- HTTP Message Type***: Request
- HTTP Method**: POST
- URI**: /protect/text\$
- Request Headers**: (empty)
- Message Body**: (empty)
- Require Client Certifica...**: Checked
- Authentication**: Basic
- Target Object**: Message Body

Figure 17-5: Enabling Client Certificate for a Rule

- Save the changes.

17.2.2 TLS Mutual Authentication

DSG can be configured with trusted root CAs and/or the individual client machine certificates for the machines that will be allowed to connect to DSG's REST API TLS tunnel.

Client machines that fail to offer a valid client certificate will not be able to connect to DSG's REST API TLS ports.

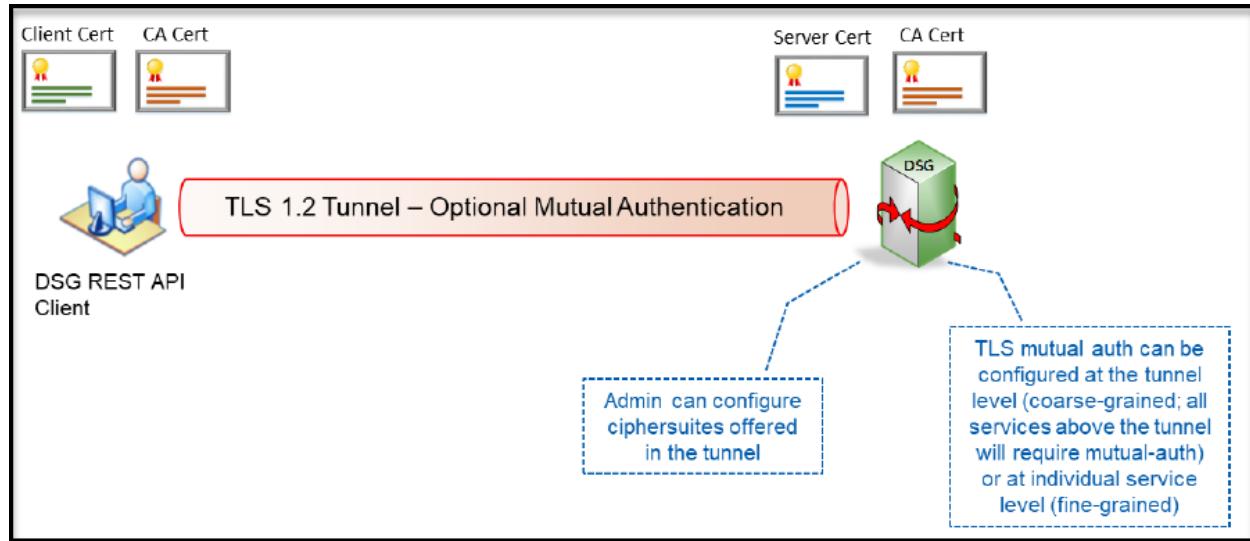


Figure 17-6: REST API TLS Mutual Authentication

When communicating with DSG, you can add an additional layer of security by requesting the HTTP client to present a client certificate. You can do so by enabling mutual authentication in DSG. The certificate presented by the client must be trusted by DSG in order for the TLS connection to be successfully established.

Before rolling out to production, in the testing environment, you can use DSG to generate the client key and certificate. The client key and certificate are self-signed by DSG using a self-generated CA.

17.2.2.1 Enabling Mutual Authentication

DSG supports TLS mutual authentication and you can use steps in this section to enable it.

Before you begin

Ensure that the following prerequisites are completed in the given order.

1. Ensure that you generate custom CA certificates and keys, for example, a certificate file `ca_test.crt`, a key file `ca_test.key`, a pem file `ca_test.pem` and upload it from the *Certificate/Key Material* screen on the DSG.
2. Ensure that you generate custom server certificates and keys, for example, a certificate file `server_test.crt`, a key file `server_test.key`, a pem file `server_test.pem`, and sign it with the CA certificates generated in step 1. After creating the certificates and keys, upload it from the *Certificate/Key Material* screen on the DSG.
3. Ensure that you generate custom client certificates and keys, for example, a certificate file `client_test.crt`, a key file `client_test.key`, a pem file `client_test.pem`, and sign it with the CA certificates generated in step 1. You must use the client certificate while sending a request to the DSG.

► To enable Mutual Authentication:

1. Configuring Tunnel to enable Mutual Authentication.
 - a. On the ESA Web UI, navigate to the *Tunnels* screen and click the **HTTP tunnel** that you want to edit.
 - b. In **TLS Mutual Authentication**, set the value as **CERT_OPTIONAL**.
 - c. In **CA Certificates**, provide the absolute path of the `ca_new.pem` certificate.
2. Configuring rule to enable Mutual Authentication.
 - a. In the extract rule, ensure that you select the **Require Client Certificate** check box.

17.3 Protecting an XML Document through DSG REST API

Let's review a use case where a client requests protection of certain sensitive information within an XML document through DSG's REST API.

While this use case describes fine-grained protection of an XML document, one can easily translate this into other structured data formats such as CSV, JSON etc. or even unstructured data formats.

As a precursor to understanding DSG RuleSet configuration, let's first review the REST API input and the expected output. The following snippet shows sample REST API request and response messages. The client sends an HTTP POST request message that carries an XML document in it. The expectation is that the content of `<Person><Name>...</Name></Person>` XML hierarchy be protected at word boundaries. DSG responds back with an updated XML document wherein the specified sensitive data been substituted with tokens.

Request:

```
POST /protect/xml/xpath HTTP/1.1
Host: restapi
Content-Length: 85
Content-Type: application/xml
<Person>
  <Title>Mr.</Title>
  <Name>Joe Smith</Name>
  <Gender>Male</Gender>
</Person>
```

Response:

```
HTTP/1.1 200 OK
Transfer-Encoding: chunked
Content-Type: application/xml
Server: Protegity Cloud Gateway 1.0.0.170
```

```
<Person>
  <Title>Mr.</Title>
  <Name>nM9M 4NFuRl9</Name>
  <Gender>Male</Gender>
```

To produce the API output shown above, DSG is configured with a RuleSet object ahead of time. The RuleSet object is tied to a REST API URL (/protect/xml/xpath in this example). As mentioned earlier, a RuleSet object is a collection of all the rules responsible for handing requests arriving on a specific URL. The handling of this request is decomposed in four cascading steps as shown in the following figure.

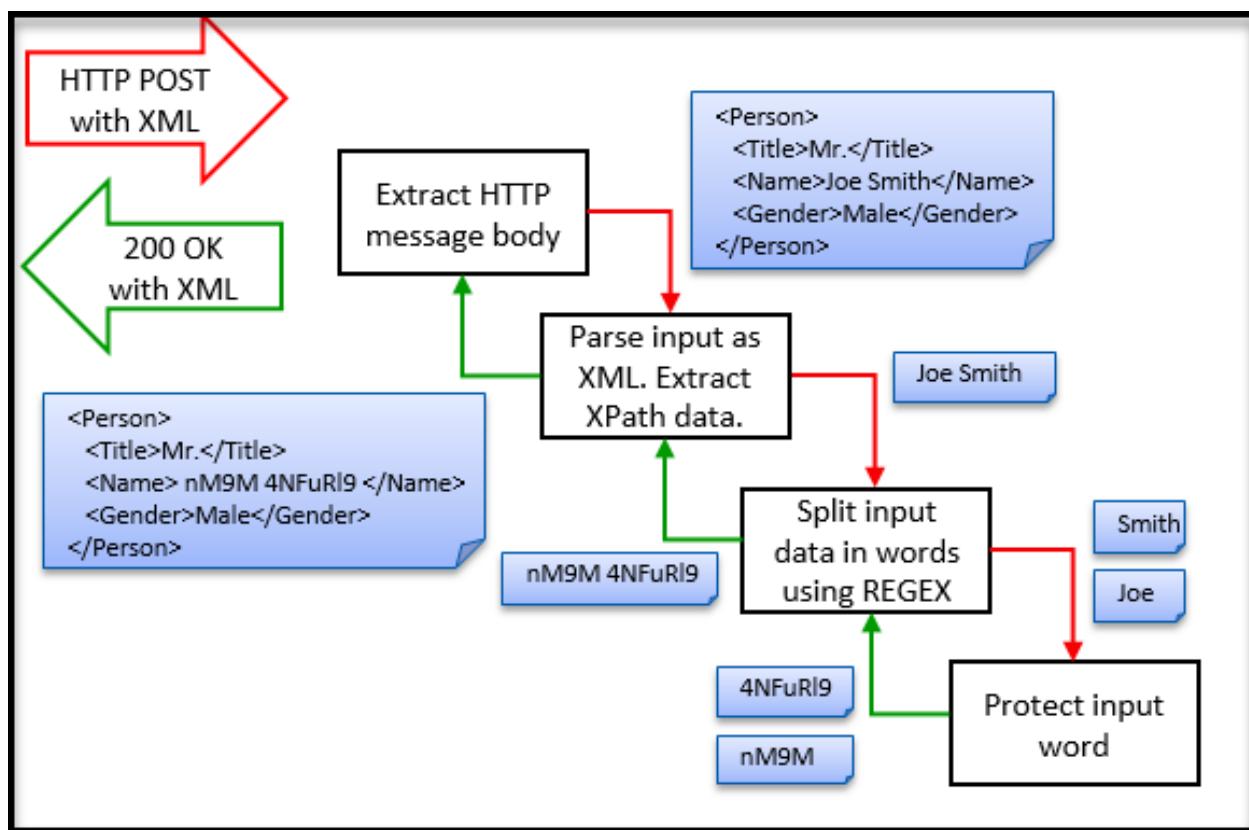


Figure 17-7: RuleSet Example

Step 1: Extract body of the HTTP request message. The extracted body content will be the entire XML document. The extracted output of this Rule will be fed to all its children sequentially. In this example, there is only one child of this Rule node.

Step 2: Parse the input as an XML document such that an XPath can be evaluated on it to find position offsets of the sensitive data content.

Note: One may choose to treat the XML document as a flat text file and run a REGEX on it instead and get the same results. The choice depends on studying the XML and where all sensitive data lies and which mechanism will yield more accurate, maintainable and high performant results.

Step 3: Split the extracted data from the previous rule into words. This will be done by running a simple REGEX on the input. Each word will be fed into children rule nodes of this rule one by one. In this use case there is only one child to this Rule.

Step 4: Protect input content. Since this Rule is a leaf node (a node without any children), return resulting ciphertext or token to the parent.

At the end of Step 4, the RuleSet traversal stack will unwind and each branch Rule node will reverse its previous action such that the overall data can be brought back to its original format. Going back in the reverse direction, Step 4 will return tokens to Step 3 which will concatenate them together into a string. Step 2 will substitute the string yielded back from Step 3 into the original XML document in place of the original plaintext string pointed at by the configured XPath. Step 1 that was originally responsible for extracting body of the HTTP request will now replace what has been extracted with the modified XML document. A layer of platform logic outside of RuleSet tree execution will create an HTTP response message which will convey the modified XML document to the client.

Let us translate the request handling design described above into real DSG configuration. The figures from Step 2 Rule Node Configuration to [Postman Client Example](#) depict Rule nodes creation for Step 1 through Step 4 in the ESA Web UI under the Cloud Gateway section.

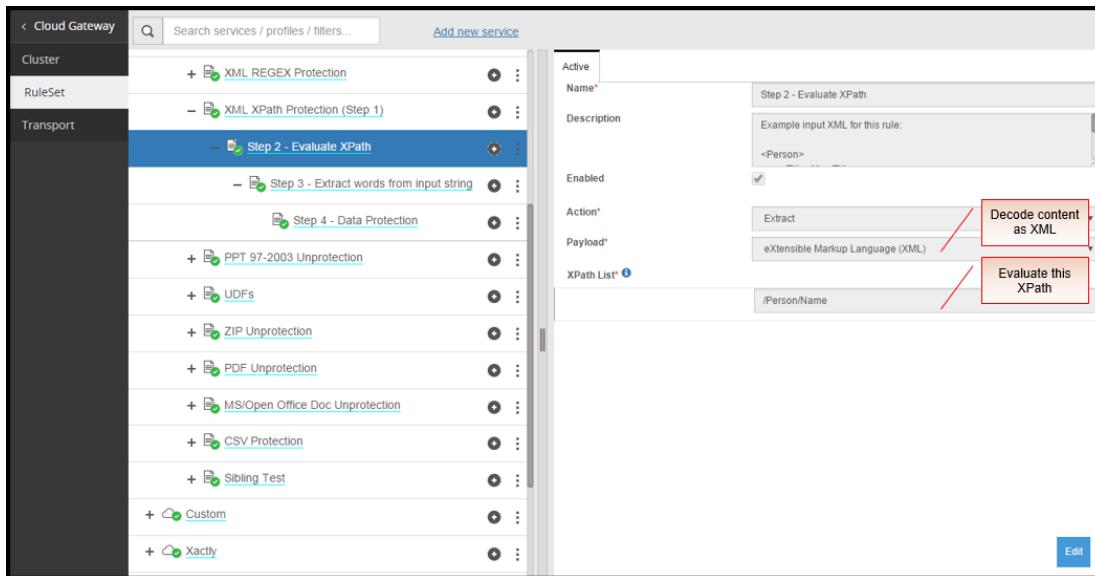


Figure 17-8: Step 2 Rule Node Configuration

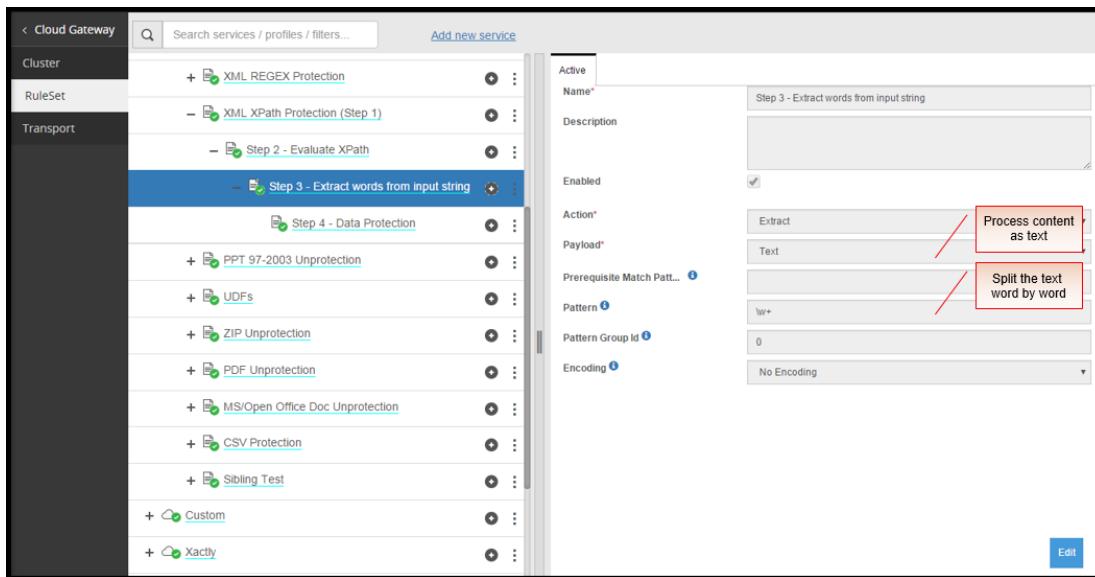


Figure 17-9: Step 3 Rule Node Configuration

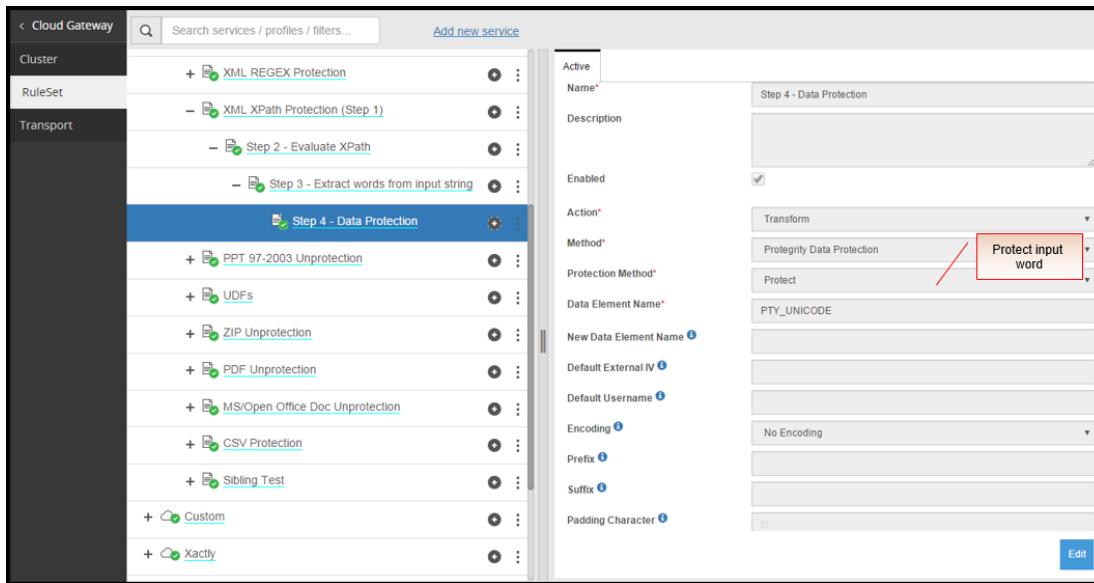


Figure 17-10: Step 4 Rule Node Configuration

17.4 Java Client using Native java.net.HttpURLConnection

This section describes REST API example with JAVA client.

```

import java.io.BufferedReader;
import java.io.DataOutputStream;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;

/*
 * DSG REST API Client Example:
 *
 * This example simulates a DSG REST API client. The example uses Java native
 * HttpURLConnection utility for sending an HTTP POST request to a configured
 * REST API end-point URL in DSG ("http://restapi/protect/xml/regex" or
 * "http://restapi/protect/xml/xpath").
 *
 * The example sends an XML document with sensitive content in plain-text and
 * expects to receive an HTTP response (200) where such content has been
 * substituted by cipher-text or tokens. In order to support this example, the
 * DSG has been configured with a corresponding REST API RuleSet that protects
 * information contained in <Name>...</Name> XML tag.
 *
 * The DSG REST API consumes the entire XML document, finds the sensitive
 * information in it either based on a configured REGEX or XPath, and responds
 * back with a modified XML document where the sensitive information has been
 * protected. No other part of the input document is modified.
 *
 * While this example demonstrates XML as the subject content, DSG supports a
 * variety of content types including all textual formats (XML, JSON, HTML, CSV,
 * JS, HTML etc.) as well as complex formats such as XLS(X), PPT(X), DOC(X),
 * TSV, MHT, ZIP and PDF.
 */

public class RestApiSampleHttpURLConnection {
    public static void main(String[] args) {

        // DSG's REST API end point URL - tied to REGEX or XPath based RuleSet
        String dsgRestApiUrl = "http://restapi/protect/xml/xpath";
        // String dsgRestApiUrl = "http://restapi/protect/xml/regex";

        HttpURLConnection conn = null;
        try {
            // Create connection
            URL url = new URL(dsgRestApiUrl);

```

```

conn = (HttpURLConnection)url.openConnection();

// Set headers and body
conn.setRequestMethod("POST");
conn.addRequestProperty("Content-Type", "application/xml");
conn.setDoOutput(true);

// Send request
DataOutputStream wr = new DataOutputStream (conn.getOutputStream());
String inXml = "<Person>
    +      "<Title>Mr.</Title>"
    +      "<Name>Joe Smith</Name>"
    +      "<Gender>Male</Gender>"
    +  "</Person>";
wr.writeBytes(inXml);
wr.close();
System.out.println("REQUEST: \n" + "POST " + dsgRestApiUrl + "\n" + inXml);

// Receive response line by line
BufferedReader in = new BufferedReader(
    new InputStreamReader(conn.getInputStream()));
String inputLine;
StringBuffer response = new StringBuffer();
while ((inputLine = in.readLine()) != null) {
    response.append(inputLine);
}
in.close();
System.out.println("RESPONSE: \n" + response.toString());
catch (Exception e) {
e.printStackTrace();
}

```

17.5 Python Client

This section showcases REST API example with Python client.

```
#!/usr/bin/python
import http.client
reqBody = "<Person>
            <Title>Mr.</Title>
            <Name>Joe Smith</Name>
            <Gender>Male</Gender>
        </Person>"
headers = {'Content-type': 'application/xml'}
conn = http.client.HTTPConnection('restapi')
conn.request('POST', '/protect/xml/xpath', reqBody, headers)
print(conn.getresponse().read())
```

17.6 Scala Client using Apache HTTP Client

This section showcases REST API example with SCALA client.

```
import org.apache.http.client.methods.HttpPost
import org.apache.http.impl.client.DefaultHttpClient
import org.apache.http.entity.StringEntity
import org.apache.http.util.EntityUtils;

object RestApiClient {
    def main(args: Array[String]): Unit = {
        val post = new HttpPost("http://restapi/protect/xml/xpath")
        post.addHeader("Content-Type", "application/xml")
        val inXml = "<Person>" +
                    "<Title>Mr.</Title>" +
                    "<Name>Joe Smith</Name>" +
                    "<Gender>Male</Gender>" +
                    "</Person>";
    }
}
```

```

    val client = new DefaultHttpClient
    val params = client.getParams
    post.setEntity(new StringEntity(inXml))
    val response = client.execute(post)
    val rspBodyString = EntityUtils.toString(response.getEntity)
    println("RESPONSE: \n" + rspBodyString)
}
}

```

17.7 Postman (Chrome Plugin) Client

This section describes a REST API example with the Postman client.

The screenshot shows the Postman interface in a browser window. The URL is `http://restapi/protect/xml/xpath`. The method is set to `POST`. The `Content-Type` is `application/xml`. The `Header` is `Value`. The `Body` tab is selected, showing the XML document:

```

1 <Person>
2   <Title>Mr.</Title>
3   <Name>Joe Smith</Name>
4   <Gender>Male</Gender>
5 </Person>

```

A red arrow points from the number 2 in the XML code to the `Value` field in the Header section. A callout box states: "Example REST API end-point URL configured in PCG for protecting XML documents based on XPath". Another red arrow points from the number 3 in the XML code to the `Value` field in the Body section. A callout box states: "Example XML document sent in HTTP POST request. XPath configured in PCG: Person/Name".

The response section shows the status as `200 OK` with a time of `124 ms`. The `Body` tab is selected, displaying the XML response:

```

1 <Person>
2   <Title>Mr.</Title>
3   <Name>nM 4NFuR19</Name>
4   <Gender>Male</Gender>
5 </Person>

```

A red arrow points from the number 2 in the XML response to the `Value` field in the Body section. A callout box states: "HTTP 200 response from PCG with word by word protection of information contained in Person/Name XPath".

Figure 17-11: Postman Client Example

Chapter 18

User Defined Functions (UDF)

[18.1 User Defined Functions](#)

[18.2 RuleSet Tree Recursion and Generators](#)

[18.3 User Defined Variables in the UDFs](#)

[18.4 Passing input arguments in UDFs](#)

[18.5 Advanced Rule Settings in UDFs](#)

[18.6 Python code listing of Sample UDFs](#)

[18.7 Blocked Modules and Methods in UDF](#)

[18.8 UDFs in the Web UI](#)

[18.9 Supported Libraries](#)

The Data Security Gateway (DSG) UDFs are API hooks that allow users to insert certain kinds of custom program logic in the DSG data processing flow.

Note:

From DSG 3.0.0.0, the Python version is upgraded to python 3. The UDFs written in Python v2.7 will not be compatible with Python v3.10. To migrate the UDFs from python 2 to python 3, refer to the section [Appendix I: Migrating the UDFs to Python 3](#).

While DSGs built-in CoP building blocks (e.g. standard protocol codecs included in the base DSG delivery) allow configuration-driven handling for most data security use cases seen in typical web and networking protocols, the DSG UDF capability is designed for addressing unique customer requirements that are otherwise not possible to address through configuration only. Such requirements may include extracting relevant data from proprietary application layer protocols and payload formats or altering data in some custom way.

Note: The DSG UDF mechanism is designed for customizations and extensibility of the DSG product deployments in the field. Any UDF code (even if developed with assistance from Protegility) is part of the customer-specific deployment and is not a part of the base DSG product delivery from Protegility. Customers are responsible for the functionality, quality, and on-going maintenance of their DSG UDF code.

This section describes the kinds of UDFs currently available in the DSG, where they fit within the DSG architecture, their interfaces with the main program, and rules for implementing and coding them.

18.1 User Defined Functions

The Extraction (Extract) and Transformation (Transform) rule types are responsible for actual data processing and are therefore enabled with UDF functionality.

The concept of UDFs is not new. They are prevalent in the RDBMS world as a means of inserting custom logic in database queries or stored procedures. As opposed to full blown APIs with strict client/server semantics, UDFs provide a somewhat

contained mechanism for inserting a small piece of logic within the context of an existing execution flow. UDFs are basically call-backs, which mean that they must comply with the calling program's interface and typically must not negatively affect the overall execution flow in terms of their added latency.

The DSG Extraction and Transformation UDFs are user-written pieces of logic that must comply with the DSG Rules Engine interfaces such that control and data can be handed back and forth between the main program and the UDF. However, beyond complying with the DSG's interface, UDF writers have complete freedom in what they want to achieve within the UDF.

The following figure shows an example RuleSet tree with an Extraction and a Transformation Rule object that are defined as UDFs (code listings in Appendix A). In the example, the Extraction UDF performs word by word extraction of input data while the Transformation UDF toggles alphabet cases for each word passed into it.

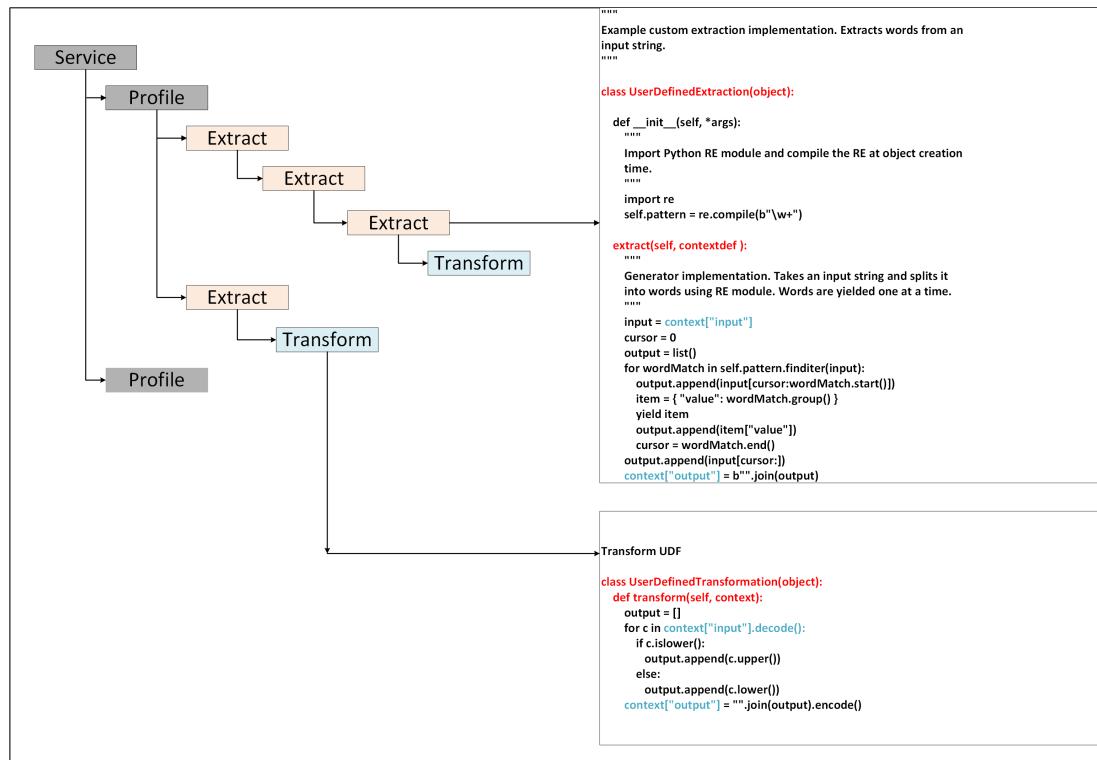


Figure 18-1: Extraction and Transformation UDFs implemented as API hooks

18.2 RuleSet Tree Recursion and Generators

This section describes how Ruleset trees must be defined for UDFs.

While the DSG Rules Engine is responsible for executing RuleSet tree, the actual DSG data processing behavior is nothing but the outcome of tree recursion where Rule behaviors are executed in the order they are laid out in the tree. Since the design of RuleSet tree is completely configurable, this approach has generally been referred to as Configuration-over-Programming (CoP).

Extraction rules are branch nodes responsible for mining data, whereas, Transformation rules are leaf nodes responsible for manipulating data. To achieve loose coupling between Rule objects, lazy searches over data and simplicity of programming, Extraction rules are implemented as Generators. Currently, the DSG UDFs are programmable in Python (3.10), which means that Extraction UDFs are written with Python yield keyword. This allows Extraction UDFs to be performance efficient (by way of lazy searching) while at the same time supporting an iterator interface without returning an iterator as a data structure collection. The following figure shows how an Extraction rule works as a Generator.

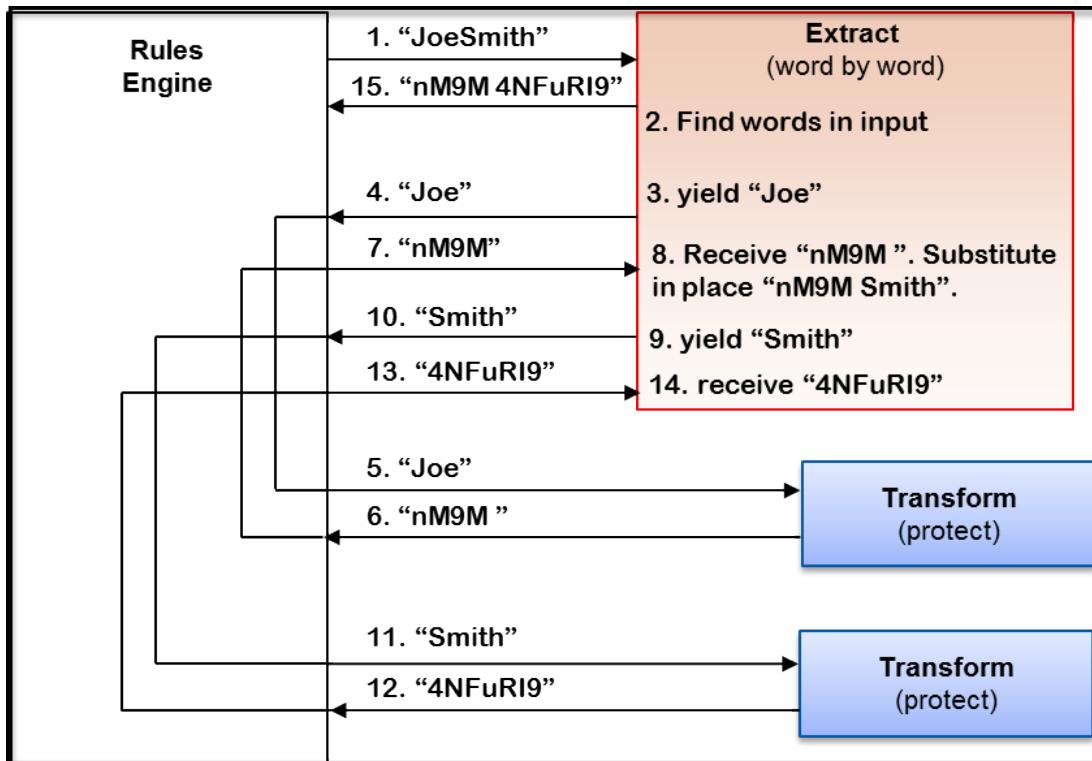


Figure 18-2: Extraction Rules operate as Generators

Transformation UDFs require a simple Python class and typically only one method to be implemented. Users implement a Python class called UserDefinedTransformation and implement a transform method in it. The transform method inputs a dictionary (named context in the following example). This dictionary uses the following two keys:

context[“input”] – Data input into UDF

context[“output”] – Data output from UDF (transformed in some way)

Note: The input and the output data must be in bytes.

```

class UserDefinedTransformation(object):
    def transform(self, context):
        input = context["input"]
        # Transform input in some way and return it in output
        context["output"] = output
  
```

18.2.1 Implementing an Extraction UDF

This section explains about how the extraction UDF is defined in form of a sample UDF.

Extraction UDF writers implement a Python class called UserDefinedExtraction with an Extract method in it. The **Extract** method must be implemented as a Python generator. Similar to Transformation UDFs, Extraction UDFs input a dictionary with *input* and *output* keys. In addition, Extraction UDFs use another dictionary for returning Generator output (named item in the following example) with **value** key. Code listing with comments in the following snippet describe the interfaces with the calling program.

```

class UserDefinedExtraction(object):
    def extract(self, context):
        input = context["input"]
        # Extract desired pieces of data from input
        # Return/yield extracted pieces (one by one) to caller for .... :
        # Populate item dict. with value key in it with each extracted piece of
        data
  
```

```

item = { "value": extractedData }
# Yield extracted pieces of data. They will be passed on to Transformation
rules yield item
# Transformed data will be available in item dictionary with value key
transformedData = item["value"]
# Transformed data is assembled back in output and returned to caller
context["output"] = output

```

18.3 User Defined Variables in the UDFs

This section explains how the sample UDF defined user defined variables.

The Extraction and Transformation UDFs allow users to define their own variables that are maintained throughout the scope of RuleSet execution. This is useful in passing information across different UDFs e.g. setting a variable in one UDF and retrieving it in another UDF. A specific key called cookies has been reserved in the context dictionary for this purpose.

For example, users may use the cookies key to set their own dictionary of parameters and retrieve in a UDF called subsequently.

```

context["cookies"] = { "customAuthCode": authCode }
authCode = context["cookies"]["customAuthCode"]

```

18.4 Passing input arguments in UDFs

The Transformation and Extraction UDF classes allow users to pass in a variable number of statically configured input arguments in their `__init__()` method as shown in the following screenshot.

```

Source code* ⓘ
class UserDefinedExtraction(object):

    def __init__(self, *args):
        """
        Import Python RE module and compile the RE at object creation
        time.
        """

        import re
        self.pattern = re.compile(b"\w+")

    def extract(self, context):
        """
        Generator implementation. Takes an input string and splits it
        into words using RE module. Words are yielded one at a time.
        """

Initialization Arguments ⓘ
"test_string", 99

```

Figure 18-3: Passing input arguments in UDF

18.5 Advanced Rule Settings in UDFs

The `gateway.json` file includes a configuration where vulnerable methods and modules are blocked from being imported as part of the User Defined Extract and User Defined Transform rule. This default behavior can be overruled by setting the `Rule Advanced Settings` parameter.

In the following example *source code*, the code requests to import the *os* module. This module is part of the default blocked modules in the *gateway.json* file. If as part of the UDF rule configuration, it is required that the *os* module be unblocked, then the *Rule Advanced Settings* parameter must be set as shown in the figure.

For more information about the blocked methods and modules, refer to the section [Blocked Modules and Methods in UDF](#).

Note:
Currently, methods cannot be overridden using Advanced settings.

Payload*	User Defined Extraction
Programming Language	Python
Source code* ⓘ	<pre>class UserDefinedExtraction(object): def __init__(self, *args): """ Import Python RE module and compile the RE at object creation time. """ import re self.pattern = re.compile(b"\w+") def extract(self, context): """ Generator implementation. Takes an input string and splits it into words using RE module. Words are yielded one at a time. """ </pre>
Initialization Arguments ⓘ	
Rule Advanced Settings ⓘ	{"override_blocked_modules":["os"]}

Figure 18-4: Passing input arguments in UDF

18.6 Python code listing of Sample UDFs

This section provides a python code listing of sample UDFs.

```
"""
Example custom extraction implementation. Extracts words from an
input string.
"""

class UserDefinedExtraction(object):

    def __init__(self, *args):
        """
        Import Python RE module and compile the RE at object creation
        time.

```

```

"""
import re
self.pattern = re.compile(b"\w+")

def extract(self, context):
    """
    Generator implementation. Takes an input string and splits it
    into words using RE module. Words are yielded one at a time.
    """
    input = context["input"]
    cursor = 0
    output = list()
    for wordMatch in self.pattern.finditer(input):
        output.append(input[cursor:wordMatch.start()])
        item = { "value": wordMatch.group() }
        yield item
        output.append(item["value"])
        cursor = wordMatch.end()
    output.append(input[cursor:])
    context["output"] = b"".join(output)

"""

Custom Transformation UDF: Toggles alphabet cases.

"""

class UserDefinedTransformation(object):
    def transform(self, context):
        output = []
        for c in context["input"].decode():
            if c.islower():
                output.append(c.upper())
            else:
                output.append(c.lower())
        context["output"] = "".join(output).encode()

```

18.7 Blocked Modules and Methods in UDF

This section provides the snapshot of the *globalUDFSettings* key and the blocked modules and methods as the values.

The *gateway.json* file includes the *globalUDFSettings* parameter. To access the *gateway.json* file, navigate to **Settings > System > Files**, and under the *Cloud Gateway - Settings* area, access the *gateway.json* file.

Caution:

As part of security enhancements, the vulnerable modules and methods are added as values in the *globalUDFSettings* key. It is recommended that the default blocked modules and methods are not removed from the list.

You can add any default Python modules and method to the list.

Note:

Currently, methods cannot be overridden using Advanced settings.

```

"globalUDFSettings": {
    "blocked_methods": [
        "eval",
        "exec",
        "dir",
        "__import__",
        "memoryview"
    ],
    "blocked_modules": [
        "pip",
        "install",
        "commands",

```



```
        "subprocess",
        "popen2",
        "sys",
        "os",
        "platform",
        "signal",
        "asyncio"
    ]
}
```

18.8 UDFs in the Web UI

This section shows how the UDFs appear in the Web UI.

The screenshot shows the configuration interface for a User Defined Function (UDF) named 'User Defined Extraction'. The interface is divided into several sections:

- Payload***: A dropdown menu set to 'User Defined Extraction'.
- Programming Language**: A dropdown menu set to 'Python'.
- Source code* ⓘ**: A code editor containing Python source code for a custom extraction implementation. The code defines a class 'UserDefinedExtraction' that imports the 're' module and compiles a regular expression pattern to extract words from an input string.

```
"""
Example custom extraction implementation. Extracts words from an
input string.
"""

class UserDefinedExtraction(object):

    def __init__(self, *args):
        """
        Import Python RE module and compile the RE at object creation
        time.
        """

        import re
        self.pattern = re.compile(b"\w+")

```

- Initialization Arguments ⓘ**: A section for defining initialization arguments, currently empty.
- Rule Advanced Settings ⓘ**: A section for rule advanced settings, currently empty.

A blue 'Edit' button is located at the bottom right of the code editor area.

Figure 18-5: Extraction UDF in Web UI

Action*: Transform

Method*: User Defined Transformation

Programming Language: Python

Source Code* ⓘ

```
class UserDefinedTransformation(object):
    def transform(self, context):
        output = []
        for c in context["input"].decode():
            if c.islower():
                output.append(c.upper())
            else:
                output.append(c.lower())
        context["output"] = "".join(output).encode()
```

Initialization Arguments ⓘ

Rule Advanced Settings ⓘ

Figure 18-6: Transformation UDF in Web UI

18.9 Supported Libraries

This section lists the libraries that are installed as part of the DSG installation, apart from the default Python libraries, that can be used while creating custom UDFs.

Table 18-1: Supported Libraries

Library Name	Version
boto3	1.7.23
botocore	1.10.84
jsonschema	2.6.0
kafka-python	2.0.2
lxml	4.2.1
paramiko	2.4.1
pika	0.11.2
PyJWT	1.7.1

Chapter 19

Appendix A: Xactly

[19.1 DSG with Xactly SaaS](#)

[19.2 Supported Modules](#)

[19.3 Supported Fields](#)

[19.4 Supported Profiles](#)

[19.5 Known Issues and Limitations](#)

This section provides information about how Data Security Gateway (DSG) is used to support Xactly SaaS.

19.1 DSG with Xactly SaaS

A diagrammatic representation of the Xactly and DSG architecture is provided in this section.

The integration between DSG and the Xactly SaaS is as shown in the following image.

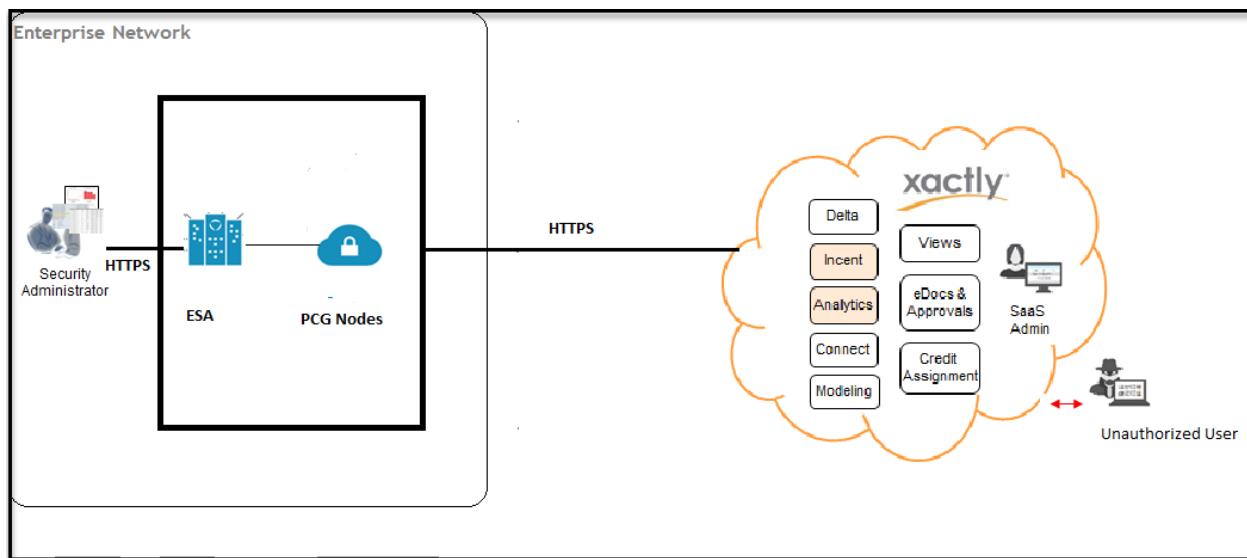


Figure 19-1: Xactly integration with DSG

19.2 Supported Modules

The integration between DSG and the Xactly SaaS is as shown in the following image. The Incent and Analytics module for Xactly are supported in this release.

Incent Module

In Xactly, the Incent module lets you manage sales performance and incentive compensation. This includes real time visibility across the organization, easier compensation calculations, reduced costs related to sales compensation, and ability to perform audit trails for transactions.

Analytics Module

In Xactly, the Analytics module lets you view the Sales Compensation Management (SCM) metrics in a unified interface. It also provides the enhanced decision making capabilities due to ad hoc analysis and pre built SCM analytic content. This helps in better understanding of sales made, improved alignment between finance and sales teams, and boost sales and financial performance.

19.3 Supported Fields

The fields supported by DSG for Xactly modules are listed in this section.

The following Xactly fields are supported in this release of DSG.

Table 19-1: Xactly Supported Fields

Field Name	Type of Field
User Name	Standard Protected Fields
Person Name	
Customer Name	
Orders > Business Program	Optional Protected Fields
Orders > Program Sub-Segment	
Orders > Sales Rep	
Orders > Payee	
People > Organization Name	

Note: If you are using custom fields in Xactly, ensure that any rules related to them are modified based on the custom field IDs configured in your Xactly instance.

19.4 Supported Profiles

The profiles supported by DSG for Xactly modules are listed in this section.

The following table provides a list of profiles, the protected Xactly modules, and its description.

Table 19-2: Xactly Supported Profiles

Profile Name	Protected Xactly Module	Description
analytics	Analytics	Protect any sensitive data encountered in the Analytics module.
incent	Incent	Protect any sensitive data encountered in the Incent module.
incentCustomerName		Protect any instance of CustomerName encountered in the Incent module.
incentCustomField		Protect any instance of CustomField encountered in the Incent module.
incentProductName		Protect any instance of ProductName encountered in the Incent module.



Profile Name	Protected Xactly Module	Description
incentUserName		Protect every instance of UserName encountered in the Incent module.
safetyNets	Analytics and Incent	Unprotect tokens in the Analytics or Incent module.
saml	Login	Protects the Login module when SSO Login is used for logging.

19.5 Known Issues and Limitations

The known issues that are applicable to this release in context to the DSG for Xactly modules are listed in this section.

The following are the known reported issues and limitation using the Xactly profile:

Table 19-3: Known Issues and Limitations

Issue Number	DSG Known Issue	Xactly Known Issue	Description
CPG-962	Folded tokens are not unprotected in PowerPoint (PPTX) when used with the Xactly profile	n/a	If you are using the Analytics module in Xactly profile and try to export PPTX files, the PPTX rendering folds a token in two lines if the text is longer than the permissible column length. In such situations, DSG is unable to unprotect such folded tokens.
CPG-836	n/a	Unprotected PowerPoint (PPTX) file is corrupted in Xactly	When a PowerPoint (PPT) file is unprotected with payload type as text, the unprotection completes successfully, but the file is corrupted.

Chapter 20

Appendix B: Salesforce Profile

[20.1 DSG with Salesforce SaaS](#)

[20.2 Supported Fields](#)

[20.3 Supported Profiles](#)

This section provides information about how Data Security Gateway (DSG) is used to support a module in Salesforce SaaS.

20.1 DSG with Salesforce SaaS

A diagrammatic representation of the Salesforce and DSG architecture is provided in this section.

The integration between DSG and the Salesforce SaaS is as shown in the following image.

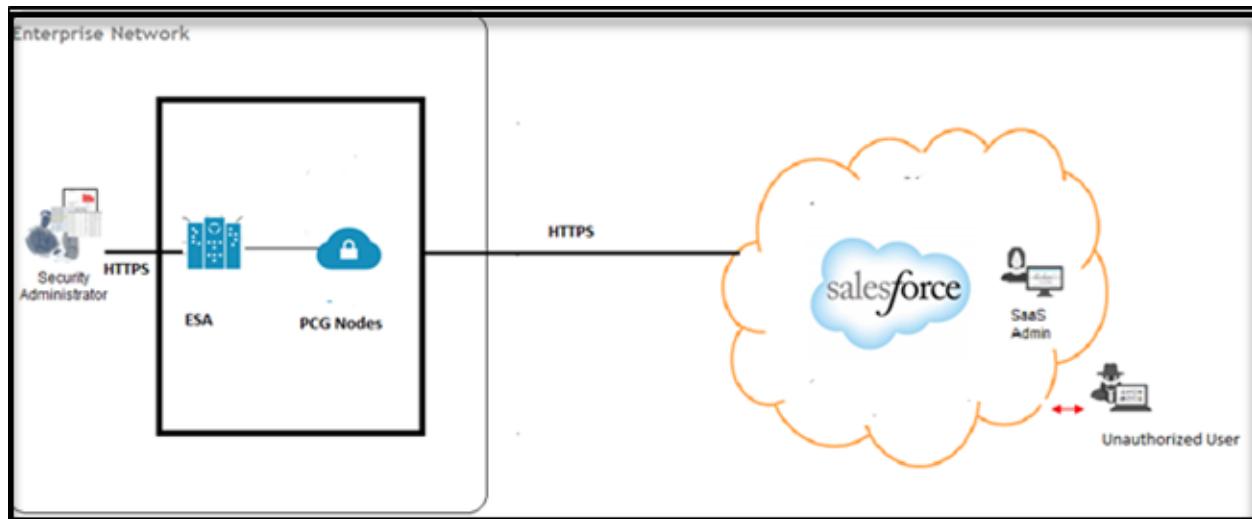


Figure 20-1: Salesforce integration with DSG

20.2 Supported Fields

The fields supported by DSG for Salesforce modules are listed in this section.

The following Salesforce fields are supported in this release of DSG.

Table 20-1: SalesForce Supported Fields

Field Name	Type of Field
Account Name	Standard Protected Fields
Account Address	

The fields mentioned in this table are the fields that are currently available. You can create rules for more fields easily from the *RuleSet* screen as per your requirements.

20.3 Supported Profiles

The profiles supported by DSG for Salesforce modules are listed in this section.

The following table provides a list of profiles and its description.

Table 20-2: Salesforce Supported Profiles

Profile Name	Description
accountAddress	Protect any instance of accountAddress encountered in Salesforce.
accountName	Protect any instance of accountName encountered in Salesforce.
safetyNets	Unprotect tokens in any Salesforce module.

Chapter 21

Appendix C: Standard Encoding Method List

A list of standard encoding methods available for as part of Data Security Gateway (DSG) Ruleset configuration is listed in this section.

The following is a list of standard encoding methods available for as part of DSG Ruleset configuration:

Table 21-1: Standard Encoding Method List

Method	Description
ascii	English (646, us-ascii)
base64	Base64 multiline MIME conversion (the result always includes a trailing '\n')
big5	Traditional Chinese (big5-tw, csbig5)
big5hkscs	Traditional Chinese (big5-hkscs, hkscs)
bz2	Compression using bz2
cp037	English (IBM037, IBM039)
cp424	Hebrew (EBCDIC-CP-HE, IBM424)
cp437	English (437, IBM437)
cp500	Western Europe (EBCDIC-CP-BE, EBCDIC-CP-CH, IBM500)
cp720	Arabic (cp720)
cp737	Greek (cp737)
cp775	Baltic languages (IBM775)
cp850	Western Europe (850, IBM850)
cp852	Central and Eastern Europe (852, IBM852)
cp855	Bulgarian, Byelorussian, Macedonian, Russian, Serbian (855, IBM855)
cp856	Hebrew (cp856)



Method	Description
cp857	Turkish (857, IBM857)
cp858	Western Europe (858, IBM858)
cp860	Portuguese (860, IBM860)
cp861	Icelandic (861, CP-IS, IBM861)
cp862	Hebrew (862, IBM862)
cp863	Canadian (863, IBM863)
cp864	Arabic (IBM864)
cp865	Danish, Norwegian (865, IBM865)
cp866	Russian (866, IBM866)
cp869	Greek (869, CP-GR, IBM869)
cp874	Thai (cp874)
cp875	Greek (cp875)
cp932	Japanese (932, ms932, mskanji, ms-kanji)
cp949	Korean (949, ms949, uhc)
cp950	Traditional Chinese (950, ms950)
cp1006	Urdu (cp1006)
cp1026	Turkish (ibm1026)
cp1140	Western Europe (ibm1140)
cp1250	Central and Eastern Europe (windows-1250)
cp1251	Bulgarian, Byelorussian, Macedonian, Russian, Serbian (windows-1251)
cp1252	Western Europe (windows-1252)
cp1253	Greek (windows-1253)
cp1254	Turkish (windows-1254)

Method	Description
cp1255	Hebrew (windows-1255)
cp1256	Arabic (windows-1256)
cp1257	Baltic languages (windows-1257)
cp1258	Vietnamese (windows-1258)
euc_jp	Japanese (eucjp, ujis, u-jis)
euc_jis_2004	Japanese (jisx0213, eucjis2004)
euc_jisx0213	Japanese (eucjisx0213)
euc_kr	Korean (euckr, korean, ksc5601, ks_c-5601, ks_c-5601-1987, ksx1001, ks_x-1001)
gb2312	Simplified Chinese (chinese, csiso58gb231280, euc-cn, euccn, eucgb2312-cn, gb2312-1980, gb2312-80, iso-ir-58)
gbk	Unified Chinese (936, cp936, ms936)
gb18030	Unified Chinese (gb18030-2000)
hex	Hexadecimal representation conversion (two digits per byte)
hz	Simplified Chinese (hzgb, hz-gb, hz-gb-2312)
iso2022_jp	Japanese (csiso2022jp, iso2022jp, iso-2022-jp)
iso2022_jp_1	Japanese (iso2022jp-1, iso-2022-jp-1)
iso2022_jp_2	Japanese, Korean, Simplified Chinese, Western Europe, Greek (iso2022jp-2, iso-2022-jp-2)
iso2022_jp_2004	Japanese (iso2022jp-2004, iso-2022-jp-2004)
iso2022_jp_3	Japanese (iso2022jp-3, iso-2022-jp-3)
iso2022_jp_ext	Japanese (iso2022jp-ext, iso-2022-jp-ext)
iso2022_kr	Korean (csiso2022kr, iso2022kr, iso-2022-kr)
latin_1	West Europe (iso-8859-1, iso8859-1, 8859, cp819, latin, latin1, L1)
iso8859_2	Central and Eastern Europe (iso-8859-2, latin2, L2)
iso8859_3	Esperanto, Maltese (iso-8859-3, latin3, L3)

Method	Description
iso8859_4	Baltic languages (iso-8859-4, latin4, L4)
iso8859_5	Bulgarian, Byelorussian, Macedonian, Russian, Serbian (iso-8859-5, cyrillic)
iso8859_6	Arabic (iso-8859-6, arabic)
iso8859_7	Greek (iso-8859-7, greek, greek8)
iso8859_8	Hebrew (iso-8859-8, hebrew)
iso8859_9	Turkish (iso-8859-9, latin5, L5)
iso8859_10	Nordic languages (iso-8859-10, latin6, L6)
iso8859_11	Thai languages (iso-8859-11, thai)
iso8859_13	Baltic languages (iso-8859-13, latin7, L7)
iso8859_14	Celtic languages (iso-8859-14, latin8, L8)
iso8859_15	Western Europe (iso-8859-15, latin9, L9)
iso8859_16	South-Eastern Europe (iso-8859-16, latin10, L10)
johab	Korean (cp1361, ms1361)
koi8_r	Russian ()
koi8_u	Ukrainian ()
mac_cyrillic	Bulgarian, Byelorussian, Macedonian, Russian, Serbian (maccyrillic)
mac_greek	Greek (macgreek)
mac_iceland	Icelandic (maciceland)
mac_latin2	Central and Eastern Europe (maclatin2, maccentraleurope)
mac_roman	Western Europe (macroman)
mac_turkish	Turkish (macturkish)
ptcp154	Kazakh (csptcp154, pt154, cp154, cyrillic-asian)
shift_jis	Japanese (csshiftjis, shiftjis, sjis, s_jis)

Method	Description
shift_jis_2004	Japanese (shiftjis2004, sjis_2004, sjis2004)
shift_jisx0213	Japanese (shiftjisx0213, sjisx0213, s_jisx0213)
utf_32	Unicode Transformation Format (U32, utf32)
utf_32_be	Unicode Transformation Format (big endian)
utf_32_le	Unicode Transformation Format (little endian)
utf_16	Unicode Transformation Format (U16, utf16)
utf_16_be	Unicode Transformation Format (big endian BMP only)
utf_16_le	Unicode Transformation Format (little endian BMP only)
utf_7	Unicode Transformation Format (U7, unicode-1-1-utf-7)
utf_8	Unicode Transformation Format (U8, UTF, utf8)
utf_8_sig	Unicode Transformation Format (with BOM signature)
zlib	Gzip compression (zip)

Chapter 22

Appendix D: Known Limitations

[22.1 Protegility Data Protection](#)

[22.2 Hardware Sizing](#)

[22.3 Network](#)

[22.4 RuleSet Engine](#)

This section provide information about the known limitation observed in Data Security Gateway (DSG).

22.1 Protegility Data Protection

The known issues observed for the Protegility data protection related operations are listed in this section.

22.1.1 Data Element Configuration

During runtime Protegility Data Protection action rules will be validating the input against restrictions imposed by the Data Element configuration. For example, Data Element may be configured to handle a certain type or format of data (i.e. date, textual, binary, etc). Input mismatching these restrictions will result in an error.

22.1.2 Input length

Input length restrictions is subject to Data Element configuration as well. Tokenization of alphanumeric input is limited to ~4 KB while encryption is limited to ~2 GB. More information can be found in ESA and policy management documentation.

22.1.3 Null Values

Payload structures such as JSON and XML can reference empty and null values. Extraction and transformation of Null or empty values are not supported currently.

22.2 Hardware Sizing

The known issues observed for hardware sizing in DSG are listed in this section.

22.2.1 Memory

When calculating memory sizing for a DSG node one has to take into consideration the maximum payload size expected to be handled.

To determine the minimum RAM required, the max payload size should be multiplied by the number of rules and the number of CPU cores, times two. For example a 32 core CPU machine with 128 GB of RAM would be able handle execution of up to 25 rules back to back to process a 20 MB payload message for upto 200 concurrent connections.

Max Payload = (Total RAM -(3 GB + (500 MB * CPU Cores))) / (Concurrent Users * Rules Count)

3 GB represents base system operation including OS and its supporting services.

500 MB represents a worker process which is executed for each CPU Core available.

Concurrent Users represents maximum concurrent connections.

Rules Count represent rules in the rules which will be engaged during runtime to process the payload.

22.2.2 Disk space

Ideal configuration where only warning and errors are logged should be well within the minimum hardware requirements of 320 GB of disk space.

This however may not be enough for certain diagnostics scenarios where learn mode log files would be keeping a copy of every rules' input/output payload. Learn mode will shut off automatically should free disk space cross the minimum threshold of 1 GB.

22.3 Network

The known issues observed for network planning in DSG are listed in this section.

22.3.1 TLS Overhead

The DSG uses software based SSL termination. The cost of which is 10% CPU overhead relative to using a clear communication channel.

22.3.2 SFTP Protocol

The known issues observed for SFTP protocol in DSG are listed in this section.

22.3.2.1 Extended SFTP Commands

Commands such as chgrp and chown are not yet supported through the gateway.

22.3.2.2 Host Key Caching

A warning log is generated for every outbound SFTP connection. This is due to the outbound host key trust/caching list not yet persistent.

22.3.2.3 Session Negotiation

SFTP session negotiation is expected to be initiated within 10 seconds.

Client applications which queue open a SFTP connection but delay the session negotiation process may suffer connection termination due to timeout. This timeout is yet to be configurable.

22.4 RuleSet Engine

The known issues observed for the Ruleset engine in DSG are listed in this section.

22.4.1 XML Payload Extractor

Order of XML tag attributes may change, CDATA tag and closing tags may be optimized by the internal libxml library used to parse XML payload. Thus, output XML structure may be structured differently compared to the input it is sourced from.

Chapter 23

Appendix E: Supported OpenSSL Curve Names and Options

This appendix provides information about the OpenSSL curve names and options supported by Data Security Gateway (DSG).

Table 23-1: Supported Curve Names

Curve Name	Description
secp112r1	SECG/WTLS curve over a 112-bit prime field
secp112r2	SECG curve over a 112-bit prime field
secp128r1	SECG curve over a 128-bit prime field
secp128r2	SECG curve over a 128-bit prime field
secp160k1	SECG curve over a 160-bit prime field
secp160r1	SECG curve over a 160-bit prime field
secp160r2	SECG/WTLS curve over a 160-bit prime field
secp192k1	SECG curve over a 192-bit prime field
secp224k1	SECG curve over a 224-bit prime field
secp224r1	NIST/SECG curve over a 224-bit prime field
secp256k1	SECG curve over a 256-bit prime field
secp384r1	NIST/SECG curve over a 384-bit prime field
secp521r1	NIST/SECG curve over a 521-bit prime field
prime192v1	NIST/X9.62/SECG curve over a 192-bit prime field
prime192v2	X9.62 curve over a 192-bit prime field
prime192v3	X9.62 curve over a 192-bit prime field
prime239v1	X9.62 curve over a 239-bit prime field



Curve Name	Description
prime239v2	X9.62 curve over a 239-bit prime field
prime239v3	X9.62 curve over a 239-bit prime field
prime256v1	X9.62/SECG curve over a 256-bit prime field
sect113r1	SECG curve over a 113-bit binary field
sect113r2	SECG curve over a 113-bit binary field
sect131r1	SECG/WTLS curve over a 131-bit binary field
sect131r2	SECG curve over a 131-bit binary field
sect163k1	NIST/SECG/WTLS curve over a 163-bit binary field
sect163r1	SECG curve over a 163-bit binary field
sect163r2	NIST/SECG curve over a 163-bit binary field
sect193r1	SECG curve over a 193-bit binary field
sect193r2	SECG curve over a 193-bit binary field
sect233k1	NIST/SECG/WTLS curve over a 233-bit binary field
sect233r1	NIST/SECG/WTLS curve over a 233-bit binary field
sect239k1	SECG curve over a 239-bit binary field
sect283k1	NIST/SECG curve over a 283-bit binary field
sect283r1	NIST/SECG curve over a 283-bit binary field
sect409k1	NIST/SECG curve over a 409-bit binary field
sect409r1	NIST/SECG curve over a 409-bit binary field
sect571k1	NIST/SECG curve over a 571-bit binary field
sect571r1	NIST/SECG curve over a 571-bit binary field
c2pnb163v1	X9.62 curve over a 163-bit binary field
c2pnb163v2	X9.62 curve over a 163-bit binary field

Curve Name	Description
c2pnb163v3	X9.62 curve over a 163-bit binary field
c2pnb176v1	X9.62 curve over a 176-bit binary field
c2tnb191v1	X9.62 curve over a 191-bit binary field
c2tnb191v2	X9.62 curve over a 191-bit binary field
c2tnb191v3	X9.62 curve over a 191-bit binary field
c2pnb208w1	X9.62 curve over a 208-bit binary field
c2tnb239v1	X9.62 curve over a 239-bit binary field
c2tnb239v2	X9.62 curve over a 239-bit binary field
c2tnb239v3	X9.62 curve over a 239-bit binary field
c2pnb272w1	X9.62 curve over a 272-bit binary field
c2pnb304w1	X9.62 curve over a 304-bit binary field
c2tnb359v1	X9.62 curve over a 359-bit binary field
c2pnb368w1	X9.62 curve over a 368-bit binary field
c2tnb431r1	X9.62 curve over a 431-bit binary field
wap-wsg-idm-ecid-wtls1	WTLS curve over a 113-bit binary field
wap-wsg-idm-ecid-wtls3	NIST/SECG/WTLS curve over a 163-bit binary field
wap-wsg-idm-ecid-wtls4	SECG curve over a 113-bit binary field
wap-wsg-idm-ecid-wtls5	X9.62 curve over a 163-bit binary field
wap-wsg-idm-ecid-wtls6	SECG/WTLS curve over a 112-bit prime field
wap-wsg-idm-ecid-wtls7	SECG/WTLS curve over a 160-bit prime field
wap-wsg-idm-ecid-wtls8	WTLS curve over a 112-bit prime field
wap-wsg-idm-ecid-wtls9	WTLS curve over a 160-bit prime field
wap-wsg-idm-ecid-wtls10	NIST/SECG/WTLS curve over a 233-bit binary field

Curve Name	Description
wap-wsg-idm-ecid-wtls11	NIST/SECG/WTLS curve over a 233-bit binary field
wap-wsg-idm-ecid-wtls12	WTLS curve over a 224-bit prime field

Table 23-2: Supported Options

Options	Description
OP_ALL	Enables workarounds for various bugs present in other SSL implementations. This option is set by default. It does not necessarily set the same flags as OpenSSL's SSL_OP_ALL constant.
OP_NO_SSLv2	Prevents an SSLv2 connection. This option is only applicable in conjunction with PROTOCOL_SSLv3. It prevents the peers from choosing SSLv2 as the protocol version.
OP_NO_SSLv3	Prevents an SSLv3 connection. This option is only applicable in conjunction with PROTOCOL_SSLv3. It prevents the peers from choosing SSLv3 as the protocol version.
OP_NO_TLSv1	Prevents a TLSv1 connection. This option is only applicable in conjunction with PROTOCOL_SSLv3. It prevents the peers from choosing TLSv1 as the protocol version.
OP_NO_TLSv1_1	Prevents a TLSv1.1 connection. This option is only applicable in conjunction with PROTOCOL_SSLv3. It prevents the peers from choosing TLSv1.1 as the protocol version. Available only with openssl version 1.0.1+.
OP_NO_TLSv1_2	Prevents a TLSv1.2 connection. This option is only applicable in conjunction with PROTOCOL_SSLv3. It prevents the peers from choosing TLSv1.2 as the protocol version. Available only with openssl version 1.0.1+.
OP_CIPHER_SERVER_PREFERENCE	Use the server's cipher ordering preference, rather than the client's. This option has no effect on client sockets and SSLv2 server sockets.
OP_SINGLE_DH_USE	Prevents re-use of the same DH key for distinct SSL sessions. This improves forward secrecy but requires more computational resources. This option only applies to server sockets.
OP_SINGLE_ECDH_USE	Prevents re-use of the same ECDH key for distinct SSL sessions. This improves forward secrecy but requires more computational resources. This option only applies to server sockets.
OP_NO_COMPRESSION	Disable compression on the SSL channel. This is useful if the application protocol supports its own compression scheme. This option is only available with OpenSSL 1.0.0 and later



Chapter 24

Appendix F: Troubleshooting Data Security Gateway (DSG)

24.1 Auditing and Logging

24.2 Best Practices for Auditing and Logging

This section explains the common errors, permission restrictions, and problems you might encounter while working with Protegility Data Security Gateway (DSG).

Table 24-1: DSG Common Errors

Error /Problem	This may happen because...	Recovery Actions
<p>The following error appears in the browser when the SaaS is accessed through the Gateway:</p> <p><i>HTTP Response Code 599: Unknown.</i></p>	The SaaS server certificate is invalid.	<p>Perform one of the following steps:</p> <ul style="list-style-type: none"> • Ensure that the forwarding address is correct. • Add the SaaS server certificate to the gateway's trusted store.
	The system time on the DSG nodes is not in sync with the ESA.	<p>Perform one of the following steps:</p> <ul style="list-style-type: none"> • Synchronize the system time for all the DSG nodes performing the following steps. <ol style="list-style-type: none"> 1. From the CLI Manager, navigate to Tools > ESA Communication. 2. Select Use ESA's NTP to synchronize the system time of the node with ESA. • Consider using an NTP server for system time across all DSG nodes and the ESA.
	The DNS configuration might be incorrect.	<p>Perform one of the following steps:</p> <ul style="list-style-type: none"> • Verify that the DNS configuration for the DSG node is set as required. • Verify that the hostname addresses mentioned in the service configuration are accessible by the DSG node.
The SaaS web interface is not accessible through the browser. The following error appears on the browser: <i>HTTP Response Code 500: Internal Server Error.</i>	The DSG node is not configured to service the requested host name.	Verify if the Cloud Gateway profiles and services are configured to accept and serve the requested hostname.



Error /Problem	This may happen because...	Recovery Actions
The following error message appears on the client application while accessing DSG. <i><h1>404 : Not Found</h1></i>	The HTTP Extract Message rule configured on the DSG node cannot be invoked.	<p>Perform one of the following steps:</p> <ol style="list-style-type: none"> 1. Ensure that you have sent the request to the URI configured on the DSG. If the request is sent to the incorrect URI, then the request will not be processed. 2. Verify the HTTP Method in the HTTP request.
The following error message appears in the gateway logs: <i>Error ;MountCIFSTunnel ;check_for_new_files ;error checking for new files, Connection timed out. Server did not respond within timeout.</i>	The connection between the DSG and CIFS server is interrupted.	Restart the CIFS server and process the data.
A clustering error indicating that the host name is denied appears.	SSH key mismatch.	<p>Perform the following steps on all nodes.</p> <ol style="list-style-type: none"> 1. Navigate to CLI > Tools > SSH Config > Known Hosts. 2. Remove the host name or the host IP address that appears in the error message. 3. Add the same host name or the host IP address to the cluster.
Clustering TCP communication failure occurred.	The node is not accessible.	Navigate to Cloud Gateway > Cluster and verify that the node IP address is accessible.
The Learn mode is not working.	The Learn mode is not enabled.	<p>Perform one of the following steps:</p> <ul style="list-style-type: none"> • Enable the Learn Mode for the required service. • Configure the following Learn Mode settings in the service creation screen: <ul style="list-style-type: none"> • Mention the contents to be included in the includeResource and the includeContentType parameters. <p>For example, you can include the following resources and content types:</p> <pre>"includeResource": " .(css/png/gif/jpg/ico/woff/ttf/svg/eot)(? b)",</pre> <pre>"includeContentType": " bcss/image/video/svg b",</pre> • Mention the contents to be excluded in the excludeResource and the excludeContentType parameters. <p>For example, you can exclude the following resources and content types:</p>



Error /Problem	This may happen because...	Recovery Actions
		<pre>"excludeResource": " .(css png gif jpg ico woff ttf svg eot)(? b)", "excludeContentType": " b css image video svg b",</pre>
The following message appears in the log: <i>WarningPolicy;missing_host_key;Unknown ssh-rsa host key for <Host IP address>: f1b2e0bde5d34244ba104bab1ce66 f96</i>	The gateway issues an outbound request to an SFTP server.	The functionality of the DSG node is not affected. No action is required.
The following message appears in the log: <i>raise ValueError("unknown cipher")#012ValueError: unknown cipher</i>	An unsupported cipher is used in the Cipher field for SFTP Advanced Settings.	Use the list of ciphers supported for SFTP Inbound Settings. For more information about the list of supported ciphers, refer to the section SFTP Inbound Settings .
The following message appears in the gateway logs: <i>protegility-cg606 gateway.py:PCPG:20210415090809989914;18840;Error;Gateway;loadServices;restapi service cannot be associated with tunnel default_80. Tunnel default_80 is not loaded</i>	The <i>default_80</i> HTTP tunnel created for the service is disabled.	To access the service, you must enable the <i>default_80</i> HTTP tunnel. To enable the <i>default_80</i> HTTP tunnel, on the ESA Web UI, navigate to Cloud Gateway > Transport , and click the Tunnels tab. Select the <i>default_80</i> HTTP tunnel and click Edit . After the <i>default_80</i> tunnel is enabled, you must restart the gateway. On the Tunnels tab, click Deploy to All Nodes to restart the gateway.
When the data security operation request fails, the following message appears in the gateway logs : <i>protegility-cg354 gateway.py:PCPG:20210416081619774574;40456;Error;RestRequestHandler;display_error;default_443 REST API Examples 10.10.3.16 POST /protect/csv 500 Failed to send logs to log processor.</i>	The LogForwarder Service is stopped.	To start the service, login to DSG Web UI, navigate to System > Services , and start the LogForwarder Service .
When the data security operation is performed, the following message appears in the gateway logs: <i>protegility-cg591 gateway.py:PCPG:20210527114559844907;55080;Error;RestStreamRequestHandler;display_error;default_443 REST API Examples 10.242.2.3 POST /protect/csv 500 signature key and export value not loaded</i>		<p>Perform one of the following options:</p> <ol style="list-style-type: none"> 1. Ensure to perform the Deploy or Deploy to Node Groups operation from the <i>Cluster</i> screen on the ESA Web UI. <ol style="list-style-type: none"> a. Login to the ESA Web UI. b. Navigate to Cloud Gateway > Cluster. c. Select the Refresh drop down menu and click Deploy or Deploy to Node Groups. 2. If the DSG node is not in the cluster, then ensure to perform the ESA Communication.



Error /Problem	This may happen because...	Recovery Actions
<p>When you enter the invalid XPath value in the <i>XML with ToT Extract Rule</i>, the rule is disabled and the following message appears in the gateway logs :</p> <pre data-bbox="99 325 567 798">protegility-cg852 gateway.py: PCPG:20210412113311375180;9869;Error;ExtensibleMarkupLanguageToT:_init_;Invalid XPath given for XPath option in rule : xml tot message extract no profile reference protegility-cg852 gateway.py: PCPG:20210412113311375276;9869;Error;ExtensibleMarkupLanguageToT:_init_;Disabling rule xml tot message extract no profile reference due to XPath option error: Invalid expression</pre>		<p>Ensure that you enter the correct value in the XPath field and deploy the configuration to enable the rule.</p>
<p>An SSL error appears when the browser connects to the ESA Web UI.</p>	<p>The SSL handshake between the web browser and the ESA is not completed.</p>	<p>Update your web browser to the latest version (Internet Explorer, Google Chrome, or Microsoft Edge).</p> <p>Alternatively, perform the following steps.</p> <ol style="list-style-type: none"> In the Web UI, navigate to Cloud Gateway > Transport > Tunnels. Select the <i>default_443</i> tunnel. In the OpenSSLCipherList text box, check the version of SSL that is not allowed, for example, SSLv2 or SSLv3. Configure the Cipher list to allow the SSL version. Restart the DSG process. <p>Note: If the Cloud Gateway menu is not available on the ESA Web UI, then login again to the ESA Web UI.</p>
<p>The Cloud Gateway nodes are stopped after patch is installed from the ESA Web UI.</p>	<p>More than one SSH Keys is assigned to the node added in the cluster.</p>	<p>Perform the following steps.</p> <ol style="list-style-type: none"> Delete the cluster from the <i>Trusted Appliance Cluster</i> screen. In the ESA Web UI, navigate to Settings > Network > SSH > Known Hosts Delete all the host names under known hosts. Re-create a new cluster from the <i>Trusted Appliance Cluster</i> screen. Add a node to the cluster from the <i>Cluster Monitoring</i> screen.
<p>A connection timed out error occurs for an outbound request.</p>	<p>The outbound timeout limit is set to 20 seconds.</p>	<p>Perform the following steps.</p>



Error /Problem	This may happen because...	Recovery Actions
		<ol style="list-style-type: none"> Click Cloud Gateway > RuleSet. Click the service that requires edits for the outbound timeout limit. In the Outbound Transport Settings field, type the following code to increase the timeout limit. <pre>{ "connect_timeout": 30, "request_timeout": 180 }</pre>
<p>In the <i>Log Viewer</i> screen, if you try to search for a time stamp, the search fails.</p> <p>If you searched for 2016-07-18T07:41:15.429350 timestamp as seen in the table, the search fails.</p>	The search field accepts numerical values.	<p>Ensure that you remove the date and time separators from the timestamp that appears on the screen before you perform a search.</p> <p>For example, if you want to search for logs that occurred on 2016-07-18T07:41:15.429350 timestamp, you must remove the date and time separators, and then search for 20160718074115429350.</p> <p>You can also search for a subset of the given timestamp. For ex, you can search for date by searching “20160718” in the search field.</p>
<p>When two or more DSG appliances are in a cluster, the following error message appears in Forensics.</p> <p><i>The clock on at least 3 nodes is out of sync.</i></p>	The system time on some or all the nodes in the cluster is out of sync.	<p>Ensure that the system time is synced manually on each of the affected nodes in the cluster.</p> <p>Perform the following steps:</p> <ul style="list-style-type: none"> Synchronize the system time for all the DSG nodes performing the following steps. <ol style="list-style-type: none"> From the CLI Manager, navigate to Tools > ESA Communication. Select Use <i>ESA's NTP</i> to synchronize the system time of the node with ESA. Perform the same steps on all DSG nodes. Consider using an NTP server for system time across all DSG nodes and the ESA.
<p>The following error appears in the <i>Log Viewer</i> screen or the gateway.log file when working with the S3 tunnel:</p> <p><i>An error occurred (RequestTimeTooSkewed) when calling the ListObjects operation: The difference between the request time and the current time is too large.</i></p>	The system time on the DSG nodes is not in sync with the ESA.	<p>Perform one of the following steps:</p> <ul style="list-style-type: none"> Synchronize the system time for all the DSG nodes performing the following steps. <ol style="list-style-type: none"> From the CLI Manager, navigate to Tools > ESA Communication. Select Use <i>ESA's NTP</i> to synchronize the system time of the node with ESA. <p>Consider using an NTP server for system time across all DSG nodes and the ESA.</p>
The SSH session is terminated during the creation of a bond on the ethMNG interface.		Restart the session after the NIC bond on the ethMNG NIC is created.
A bond is created on both the network interfaces. While testing the fail over scenario		Log in to the popout console and provide the gateway for both the interfaces if they are in the

Error /Problem	This may happen because...	Recovery Actions
by disconnecting the network, the gateway will be lost on both the interfaces. As a result, both the network interfaces will be unreachable.		same network. After adding a gateway, check if the IP addresses are provided or acquired to the network interfaces. For more information about adding a default gateway to the Management NIC and Service NIC, refer to the section Configuring Default Gateway for Network Interfaces
The slave NICs do not have an IP assigned, but the following message appears during creating a bond: NIC Bonding is not available	The NICs might be on the DHCP mode.	Convert the NICs to Static mode.
The Web UI is not accessible after the NICs are bonded.		Reset the Network Bonding from the CLI Manager and bond the NICs again. For more information about resetting the NIC bonding, refer to the section NIC Bonding .
During binding NICs, the following message appears: Unknown Error	This might occur if the network is slow.	Restart the appliance queues using the following command: /etc/init.d/appliance-queues server restart
A bond is created (mode:1) on both the Network Interfaces using the DHCP mode. If you restart the system, the MAC address and the IP address are changed.		To check if bonds are intact after a restart on mode: 1, we recommend using Static IP for the Network Interfaces.
The Join Cloud Gateway Cluster operation fails.	The ESA administrator username, password, or both are incorrect.	Perform the following steps to mitigate the issue: <ul style="list-style-type: none">• Ensure that the correct ESA administrator username and password is provided.
If an SFTP service is configured with public key authentication and the SSH public and private key are uploaded to the DSG, the following error is observed in the <i>gateway.log</i> when the SFTP tunnel is deployed. <pre>Error;SftpService;loadUserAuthConfig;Service SFTP_tunnel. Exception not a valid RSA private key file in processing private key file /opt/ protegility/alliance/config/ resources/ sftp_ssh_outbound_push_private _key Oct 23 16:34:41 protegility- cg874 gateway.pyc: PCPG:20191023110441600085;4113 3;Error; Gateway;loadServices;service initialization /opt/ protegility/alliance/config/ services /sftptunnel/sftptunnel.json failed with message not a valid RSA private key file</pre>	The SSH private key uploaded to the DSG is a not valid RSA key.	Ensure that the SSH private key is a valid RSA key. The DSG supports only RSA keys for SFTP.
After the DSG is installed, an error appears in startup log in the <i>gateway</i> log. <pre>PCPG:20200507012905113945;2237 2;Error;Gateway;loadServices;s</pre>	The module or method used in the custom code for UDF payload includes a blocked method or module.	Ensure that the module or method is not listed in the blocked method and modules. The vulnerable blocked method and modules are configured as part of the <i>gateway.json</i> file. It is recommended that these modules or methods



Error /Problem	This may happen because...	Recovery Actions
<pre>service initialization /opt/protegility/alliance/ config/services/restapi/ <rule_name> failed with message Blocked python method '<module/method name>' called in UDF</pre>		<p>are not removed from the list as a security best practice.</p> <p>You can either remove the module or method from the blocked list, or override the blocked module at the rule level.</p> <p>Note: You can only override modules at the rule level.</p> <p>For more information about the list of blocked modules and methods, refer to the section Blocked Modules and Methods in UDF.</p> <p>For more information about overriding modules, refer to the section Advanced Rule Settings in UDFs.</p>
<p>The following error message appears in the log while trying to process a file using SFTP.</p> <pre>PCPG:20190901205649639207 ;56920;Error; StubServer ;check_auth_publickey;<ip_address:port> backend connection to <ip_address:port> has failed, [Errno 110] Connection timed out</pre>	<ul style="list-style-type: none"> Network connectivity issues SFTP server not reachable 	<p>Check the following list to mitigate the issue:</p> <ul style="list-style-type: none"> Check the network connection for the SFTP server. Check the SFTP server is reachable.
<p>When you access the Tokenization Portal, the DSG Node drop down does not display any node. All the other fields appear empty too.</p>	<p>It can occur due to following reasons:</p> <ul style="list-style-type: none"> Policy is synced across all ESA nodes in the cluster Policy is not deployed to the nodes in the cluster Nodes in the TAC are unhealthy Nodes on the Cluster screen are unhealthy 	<p>Perform the steps provided for resolving the probable reasons:</p> <p>Policy is synced across all ESA nodes in the cluster Solution: Ensure that the policy on ESA is in sync with the policy on any other ESA in the cluster.</p> <p>Policy is not deployed to the nodes in the cluster Solution: Ensure that the policy is deployed to all the DSG nodes in the cluster.</p> <p>Nodes in the TAC are unhealthy Solution: On the ESA Web UI, navigate to System > Trusted Appliances Cluster to verify the node health.</p> <p>Nodes on the Cluster screen are unhealthy Solution: On the ESA Web UI, navigate to Cloud Gateway > Cluster > Monitoring to verify the node health.</p>
<p>The following message is displayed.</p> <pre>No enabled service to process request from <IP address></pre>	<p>It can occur due to following reasons:</p> <ul style="list-style-type: none"> Hostname or IP address in a request is not mapped to any service in the Ruleset Tunnel is not loaded in the DSG Service is not loaded in the DSG 	<p>Perform the steps provided for resolving the probable reasons:</p> <p>Hostname or IP address is not mapped to any service Ensure that the hostname defined in the <code>/etc/hosts</code> file is used to raise a</p>

Error /Problem	This may happen because...	Recovery Actions
		<p>request as well as is defined in the service at the Ruleset level.</p> <p>If the hostname is not defined in the <i>hosts</i> file and IP address is used, then ensure that the service IP address is used to raise a request and is defined in the service at the Ruleset level.</p> <p>Tunnel is not loaded in the DSG</p> <p>Perform the following steps based on the protocol used:</p> <ol style="list-style-type: none"> 1. Login to the DSG CLI manager. 2. Navigate to Administration > OS Console. 3. For HTTP, SFTP, or SMTP tunnels, execute the following command. <pre>netstat -nap grep <tunnel_port></pre> <p>Check if the port mapped to the service IP is same as defined in the tunnel configuration.</p> <p>Note: If the command does not return the expected output, then navigate to the <i>/var/log</i> directory and check the <i>gateway.log</i> for any exceptions during startup.</p> <ol style="list-style-type: none"> 4. For NFS tunnel, execute the following command. <pre>df -h</pre> <p>Check if the tunnel is mounted.</p> <p>Note: If the command does not return the expected output, then navigate to the <i>/var/log</i> directory and check the <i>gateway.log</i> for any exceptions during startup.</p> <ol style="list-style-type: none"> 5. For an S3 tunnel, navigate to the <i>/var/log</i> directory and check the <i>gateway.log</i> for any exceptions during start up. <p>Service is not loaded in the DSG</p> <ol style="list-style-type: none"> 1. Login to the DSG CLI manager. 2. Navigate to Administration > OS Console. 3. Navigate to the <i>/var/log</i> directory and check the <i>gateway.log</i> for any exceptions during startup.

Error /Problem	This may happen because...	Recovery Actions
The following error messages appear in <i>gateway.json</i> . <pre>PCPG:20200630071506415715;1664 3;Error;ProtegityDataProtecti on;transformProtect ;IV can't be used with this token element (returnCode 0) filter Data Protection PCPG:20200630071506415810;1664 3;Error;ProtegityDataProtecti on;transform;Failed to transform. Rule Data Protection Err IV can't be used with this token element PCPG:20200630071506419950;1664 3;Error;RestRequestHandler;log _exception;IV can't be used with this token element (returnCode 0)</pre>	An Alpha-Numeric (0-9, a-z, A-Z) token type is used, with the case-preserving and position-preserving property enabled, in the Transform rule to transform sensitive data.	Ensure that you disable the case-preserving and position-preserving property in the Alpha-Numeric (0-9, a-z, A-Z) token type in order to use External IV property.
When CSV input with non-ASCII or Unicode data is processed using the <i>CSVextract</i> rule, the following error appears. <pre>File "rule/extract/ cCharacterSeparatedValues.pyx" , line 520, in extract UnicodeDecodeError: 'ascii' codec can't decode byte 0xc3 in position 31: ordinal not in range(128)</pre>	n/a	Ensure that the <i>Binary</i> extract rule is used before using the <i>CSVextract</i> rule.
When using SFTP service, the following error appears. <pre>Aug 30 22:10:18 DSG-1 gateway.pyc: PCPG:20200830164018867619;6534 5;Verbose; SFTPServer;_log; raise IOError(text) Aug 30 22:10:18 DSG-1 gateway.pyc: PCPG:20200830164018867666;6534 5;Verbose; SFTPServer;_log;IOError: SETSTAT unsupported</pre>	The SFTP client is trying to upload a file through the DSG to the <i>AWS Transfer for SFTP</i> server.	Check the following list to mitigate the issue. <ul style="list-style-type: none"> • Ensure that the <i>enable_setstat</i> parameter is set to <i>False</i>. • Verify that the value set for the <i>enable_setstat</i> parameter is a <i>boolean</i> value.

24.1 Auditing and Logging

For every configuration change that occurs on the DSG, such as, creation of tunnels, Rulesets, deploying of the configuration, and so on, the DSG generates an audit log. Though most of the logs are forwarded to the ESA and are visible on **Forensics**, some DSG logs serve a specific purpose and are available only on the individual DSG nodes.

24.1.1 Forensics

The log management mechanism for Protegity products forwards the logs to the Audit Store on the ESA.

The following services forwards the logs to the audit store:

- **td-agent** : It forwards the appliance logs to the Audit Store on the ESA.



- **Log Forwarder:** It forwards the data security operations-related logs, such as, protect, unprotect, and reprotect and the PEP server logs to the Audit Store on the ESA.

For more information about logs in Protegility products, refer to the [Protegility Log Forwarding Guide 9.1.0.5](#).

Note: Before you can access **Forensics**, you must configure the DSG to forward logs to the *Audit Store* on the ESA. You must also verify that the *td-agent* and the *Log Forwarder* services are running on the DSG. To verify the service status, navigate to **System > Services** on the DSG Web UI.

For more information about configuring the DSG to forward appliance logs to the audit store, refer to the section [Forwarding Appliance Logs to the Audit Store](#).

For more information about configuring *Log Forwarder* to forward the audit logs, refer to the section [Forwarding Audit Logs to the Audit Store](#).

Note: The Log Forwarder configuration can be modified in the *pepserver.cfg* file. If the *Log Forwarder mode* in the *pepserver.cfg* file is modified to *error* mode or *drop* mode, then the **Pepserver** service and the **Cloud Gateway** service should be restarted.

To restart the services, login to the DSG Web UI, navigate to **System > Services**, restart the **Pepserver** service, and then restart the **Cloud Gateway** service.

For more information about logging configuration in the *pepserver.cfg* file, refer to the section [Appendix A: PEP Server Configuration File](#) in the [Protegility Installation Guide 9.1.0.5](#).

To access the **Forensics** logs, on the ESA Web UI, navigate to **Analytics > Forensics**. The **Audit** tab displays audit logs for the following events:

- Tunnel creation, deletion, and updates
- Ruleset creation, deletion, and updates
- Certificate upload, deletion, and downloads
- Key upload
- Deploying the DSG configurations
- Configuration changes made in the **Global Settings** tab
- Data security operations, such as, protect, unprotect, and re-protect
- System logs
- PEP server logs

24.1.2 Log Viewer

The **Log Viewer** displays the aggregation of the gateway logs for all the DSG nodes in the cluster. To access the **Log Viewer** file, on the ESA Web UI, navigate to **Cloud Gateway > Log Viewer**.

Important: The gateway logs are not forwarded to the *Audit Store* on the ESA.

The **Log Viewer** file details log messages that can be used to analyze the following events:

- UDF-related compilation errors
- Transaction metrics logs

- Stack traces to debug exceptions

Note: The [gateway.log](#) file can also be forwarded to any log forwarding system, such as, a Security Information and Event Management (SIEM) system or AWS CloudWatch utility.

For more information about log forwarding, refer to the section [Forwarding logs to SIEM systems](#).

24.1.3 Audit Log Representation

The DSG has the *Log Forwarder* service that forwards the logs related to the data security operations, such as, protect, unprotect, reprotect, and the PEP server logs to the Audit Store on the ESA. The logs generated from the DSG are collected and forwarded to the Audit Store.

The Audit Store holds the logs and these log records are used in various areas, such as, Forensics, alerts, reports, dashboards, and so on. Protegility Analytics is a component which provides the interface for viewing the Forensics data from the Audit Store. When the data security operations are performed to protect the sensitive data, an aggregated audit log is generated, and displayed on the **Forensics** page on the ESA Web UI.

Before you begin:

- Ensure that the **Analytics** component is initialized on the ESA Web UI. On the ESA Web UI, you can access the logs on the **Forensics** page only after initializing the **Analytics** component.

For more information about initializing the **Analytics**, refer to the section *Initializing the Audit Store Cluster on the ESA* in the [Protegility Installation Guide 9.1.0.5](#).

To understand auditing and logging for the DSG, consider the following example that will be processed using the CSV codec to extract the sensitive data.

```
firstname,lastname,city,country
John,Smith,London,Uk
Adam,Martin,London,Uk
Jae,Crowder,London,Uk
John,Holland,Bern,Switzerland
Marcus,Smart,Paris,France
Johnson,Mickey,Ottawa,Canada
```

For more information about extracting the CSV payload, refer to the section [CSV Payload](#).

The CSV extract rule is defined to process all the rows and columns. When a request is sent, the DSG processes the request and the data is protected.

```
firstname,lastname,city,country
5HnMc6vZ,G8bcRG7J1X,SQSSyxEgBKw,ATJuBh
CMgcxkSL,dlyfZKMIt5H,SQSSyxEgBKw,ATJuBh
Iqj0jjq,RgbFVD6GnOjT,SQSSyxEgBKw,ATJuBh
5HnMc6vZ,SQtul5Lqymz0,1dC18CiY,jTFgvSyjjROCx9QZ0w
6Tz3mgUy3aD,pqDuxmLouR,49HA83v7PO,Jb3kzS8gcyk
4iILZXVL06xs,nXhtMyK6vx8,TiRDIPY1Ik5,E1c5GhObzFF
```

After the protection operation is completed, a log is generated on the **Forensics** page on the ESA Web UI. An aggregated log is generated for all the protect operations performed by the CSV codec. In versions prior to the DSG 2.6.0.0, 24 different audit records were generated for each protect operation. Logging is now enhanced on the DSG and a single log entry with the count 24 is generated for the example. A log with the count is only displayed when the protect operation is completed successfully. In case of failure, the individual audit records will be displayed on the **Forensics** page on the ESA Web UI.



The following figure illustrates the audit log representation for the protect operation.

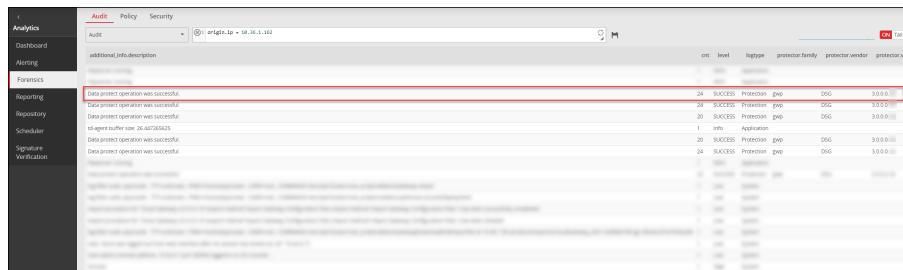


Figure 24-1: Audit Log Representation

24.2 Best Practices for Auditing and Logging

This section provides information about the best practices for auditing and logging on the DSG.

- To access the forensics, you must configure the DSG to forward logs to the *Audit Store* on the ESA. You must also verify that the *td-agent* and the *Log Forwarder* services are running on the DSG. To verify the service status, navigate to **System > Services** on the DSG Web UI.

For more information about configuring the DSG to forward appliance logs to the Audit Store on the ESA, refer to the section [Forwarding Appliance Logs to the Audit Store](#).

For more information about configuring *Log Forwarder* to forward the audit logs, refer to the section [Forwarding Audit Logs to the Audit Store](#).

- The Log Forwarder configuration can be modified in the *pepper.cfg* file. If the *Log Forwarder mode* in the *pepper.cfg* file is modified to *error* mode or *drop* mode, then the **Pepserver** service and the **Cloud Gateway** service should be restarted.

To restart the services, login to the DSG Web UI, navigate to **System > Services**, restart the **Pepserver** service, and then restart the **Cloud Gateway** service.

For more information about the logging configuration in the *pepper.cfg* file, refer to the section [Appendix A: PEP Server Configuration File](#) in the [Protegility Installation Guide 9.1.0.5](#).

- By default, the *logallcallouts* parameter is set to **no** in the *pepper.cfg* file. It is recommended to keep the default setting as is. If the *logallcallouts* parameter is set to **yes**, then duplicate logs are generated for each data security operation.

For more information about the logging configuration in the *pepper.cfg* file, refer to the section [Appendix A: PEP Server Configuration File](#) in the [Protegility Installation Guide 9.1.0.5](#).

- If the **Rsyslog** service is in stopped state on the DSG node, then the request will be processed but the audit logs will be lost. To start the service, login to the DSG Web UI, navigate to **System > Services**, and start the **Rsyslog** service.

Chapter 25

Appendix G: Enabling Selective Tunnel Loading on DSG Nodes

25.1 Adding a Label to a DSG Node for Selective Tunnel Loading

25.2 Removing a Label from a DSG Node for Selective Tunnel Loading

25.3 Adding a Label to a Tunnel for Selective Tunnel Loading

When the DSG nodes are in a cluster, generally, the ESA is responsible for pushing the Ruleset configurations to all the nodes in the cluster. In a situation where it is required that only specific tunnels are loaded on specific DSG nodes, the selective tunnel loading feature can be used.

The following figure describes how labels work with the tunnel and the DSG node definitions.

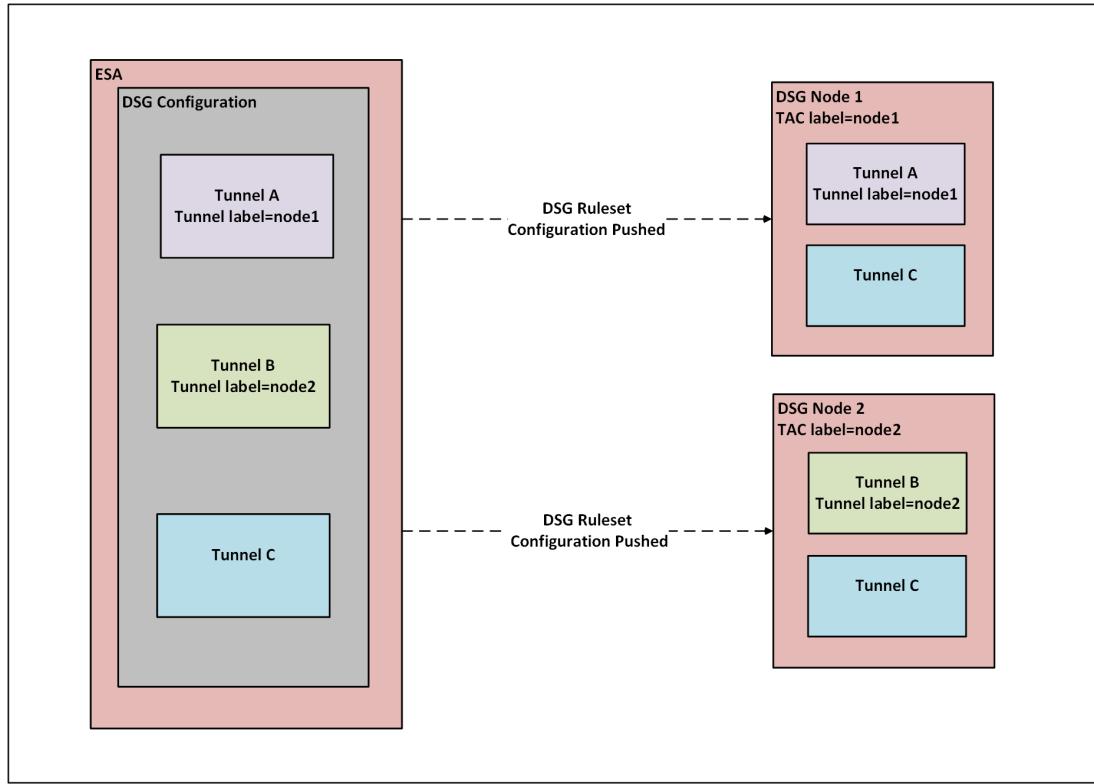


Figure 25-1: Enabling Selective Tunnel Loading

In the above figure, consider an example TAC, where exists an ESA and two DSG nodes, namely DSG Node 1 and DSG Node 2. The DSG Node 1 is an on-premise DSG, while the DSG Node 2 is a DSG on cloud. The DSG Ruleset configuration defined on the ESA includes multiple tunnel configurations, for instance, Tunnel A that must be loaded only on the DSG Node 1 and Tunnel B that must be loaded only on the DSG Node 2. The Tunnel C is common for both the DSG nodes and hence is loaded on both the nodes.

With the selective tunnel load feature, labels can be set for the tunnel and the nodes in a cluster such that only the required tunnel is loaded on the DSG node.

Perform the following steps to achieve selective tunnel loading.

1. Define the labels for each DSG node in a TAC.

For more information about adding a label to a DSG node, refer to the section [Adding a Label to a DSG Node for Selective Tunnel Loading](#).

2. Define the labels for each tunnel. This label name must match the label name defined in the DSG node label definition.

For more information about adding a label to a tunnel, refer to the section [Adding a Label to a Tunnel for Selective Tunnel Loading](#).

3. Deploy the DSG configuration from the ESA by performing the following steps.

- a. Login to the ESA Web UI.
- b. Navigate to **Cloud Gateway > Cluster**.
- c. Select the **Refresh** drop down menu and click **Deploy**.

Based on these definitions, when the DSG configuration is deployed, the Tunnel A configurations are loaded in the DSG Node 1 since the label, *node1*, defined in both the configurations is the same. Similarly, the Tunnel B configurations are loaded in the DSG Node 2 since the label, *node2*, defined in both the configurations is the same. Since no labels are defined for the Tunnel C, the Tunnel C configurations are loaded in the DSG Node 1 and DSG Node 2.

Note: The default behavior of the *Deploy* functionality does not change with the enhancements provided by Selective Tunnel Loading. In a TAC, if a configuration is pushed from the ESA, then it will be pushed to all the DSG nodes that are a part of the TAC. The configuration may include a Data Security Policy, RuleSet, Tunnels, and Services associated with a Tunnel.

25.1 Adding a Label to a DSG Node for Selective Tunnel Loading

Labels help you organize your nodes into groups or categories. By specifying a label for a node, you ensure that the node is a member of the label group. As part of enabling selective tunnel loading, the same label must be set for the DSG node in a Cluster and the Tunnel configuration. This section provides information about adding a label to the DSG node.

Before you begin

Ensure that the following prerequisites are met:

- The DSG nodes where the labels are defined must be in the same TAC.
- The TAC must be healthy.

For more information about checking cluster health, refer to the section [Monitoring tab](#).

- Ensure that the label defined for the DSG node is same as defined in the tunnel configuration.

For more information about settings the label in tunnel, refer to the section [Adding a Label to a Tunnel for Selective Tunnel Loading](#).

► To add a label to a DSG node for selective tunnel reloading:

1. Login to the DSG CLI Manager.
2. Navigate to **Tools > Clustering > Trusted Appliances Cluster**.

3. On the *Cluster Services* screen, navigate to **Node Management : Add/Remove Cluster Nodes/Information**, highlight **OK** and press **Enter**.
4. On the *Node Management* screen, navigate to **Update Cluster Information**, highlight **OK** and press **Enter**.
5. On the *Update Cluster Information* screen, navigate to **Labels**, and add the following label.

```
;dsg_onpremise
```

For example, if the NFS tunnel must be deployed only for the on-premise DSG node in a cluster, then ensure that the label parameter is set to the same label, such as *dsg_onpremise*, set for the on-premise DSG.

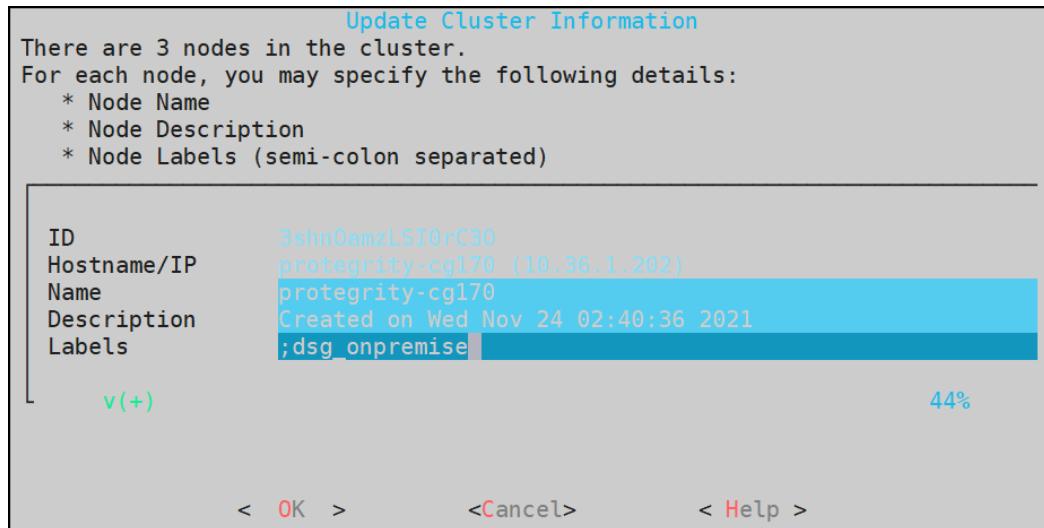


Figure 25-2: Setting label on DSG node - Cluster information

6. Highlight **OK** and press **Enter**.
7. Navigate to the *Cluster Services* screen and select **List Nodes: Show Cluster Nodes & Status** to verify if the label has been created successfully

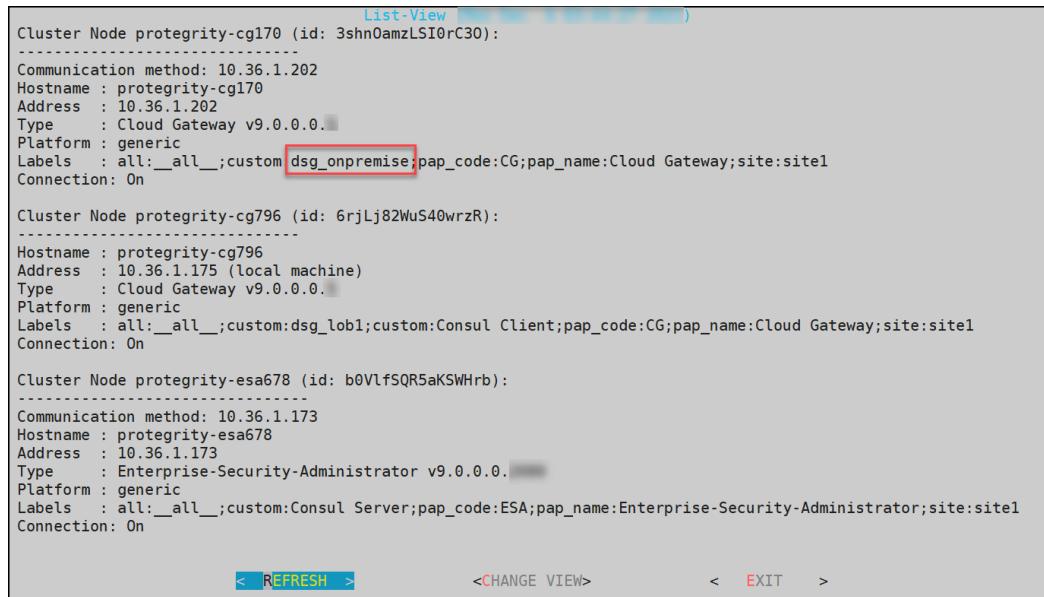


Figure 25-3: Verifying the label

25.2 Removing a Label from a DSG Node for Selective Tunnel Loading

This section describes the steps to remove a label from the DSG node for Selective Tunnel Loading. It is recommended to be cautious before removing a label from the DSG node. By removing a label for a node, you ensure that the node is not a member

of the label group. For example, if the NFS tunnel must be loaded only for the on-premise DSG node in a cluster, and a label parameter, such as `dsg_onpremise`, is removed for the on-premise DSG node, then the NFS tunnel will not be loaded on the on-premise DSG node in a cluster.

Before you begin

Ensure that the following prerequisites are met:

- The DSG nodes where the labels are removed must be in the same TAC.
- The TAC must be healthy.

For more information about checking cluster health, refer to the section [Monitoring tab](#).

► To remove a label to a DSG node for selective tunnel reloading:

1. Remove the Tunnel label from the DSG node by performing the following steps.

- a. Login to the DSG Web UI.
- b. Click **Transport > Tunnels**.
- c. Click  to edit the tunnel.
- d. Under **Advanced Settings**, remove the following key-value pair.

```
{"label": "dsg_onpremise"}
```

2. Remove the TAC label from the DSG node by performing the following steps.

- a. Login to the DSG CLI Manager.
- b. Navigate to **Tools > Clustering > Trusted Appliances Cluster**.
- c. On the *Cluster Services* screen, navigate to **Node Management : Add/Remove Cluster Nodes/Information**, highlight **OK** and press **Enter**.
- d. On the *Node Management* screen, navigate to **Update Cluster Information**, highlight **OK** and press **Enter**.
- e. On the *Update Cluster Information* screen, navigate to **Labels**, and remove the following label.

```
dsg_onpremise
```

25.3 Adding a Label to a Tunnel for Selective Tunnel Loading

As part of enabling selective tunnel reloading, the advanced settings for a tunnel configuration must be modified such that only the specific tunnel is reloaded when the matching label is found configured for a DSG node in a cluster.

Before you begin

Ensure that the following prerequisites are met:

- The DSG nodes where the labels are defined must be in the same TAC.
- The TAC must be healthy.

For more information about checking cluster health, refer to the section [Monitoring tab](#).

- Ensure that the label defined for the DSG node is same as defined in the tunnel configuration.

For more information about settings the label in the DSG node, refer to the section [Adding a Label to a DSG Node for Selective Tunnel Loading](#).

► To add a label to a Tunnel for selective tunnel reloading:

1. Login to the DSG Web UI.
2. Click **Transport > Tunnels**.
3. Click  to edit the tunnel.
4. Under **Advanced Settings**, add the following key-value pair.

```
{"label": "dsg_onpremise"}
```

For example, if the NFS tunnel must be reloaded only for the on-premise DSG node in a cluster, then ensure that the label parameter is set to the same label, such as *dsg_onpremise*, set for the on-premise DSG.

For more information about adding a label for a DSG node in a cluster, refer to the section [Adding a Label to a DSG Node for Selective Tunnel Loading](#).

Chapter 26

Appendix H: CoP Export API for deploying the CoP(Containers Only)

[26.1 Supported Authentication Methods for CoP API](#)

[26.2 API for Exporting the CoP Configurations](#)

[26.3 API for Exporting the CoP Configurations Using New Version](#)

[26.4 Scenarios for Exporting the CoP Configurations](#)

This section provides information about an API that can be used to export the CoP. The CoP technology enables a CoP Administrator to create a set of rules that will instruct the gateway on how to process data that traverses it.

For more information about CoP, refer to the section [Configuration over Programming \(CoP\) Architecture](#).

The CoP API exports the basic configurations from the ESAs *config* directory. The configurations can be retrieved from the ESA.

The following are the directories and files retrieved from the *config* directory on using the CoP API:

- Directories
 - Resources
 - Tunnels
 - Services
 - Log
- Files
 - gateway.json
 - features.json

After the required data is retrieved, the CoP API will create a *CoP* package. The *CoP* package is encrypted with the *PKCS5* standard and is *Base64* encoded.

26.1 Supported Authentication Methods for CoP API

The following authentication methods can be used to establish a connection with the ESA or DSG:

- Basic authentication with the ESA or DSG user name and password
- Client Certificate-based authentication
- Token-based authentication

For more information about accessing the ESA or DSG using these authentication mechanisms, refer to the section *Accessing REST API Resources* in the [Protegility Appliances Overview Guide 9.1.0.5](#).

DSG offers two versions of the CoP API. The following are the two base URLs:

- `https://<DSG Management IP address/ESA IP address>/exportGatewayConfigFiles`

This is the existing version of the API

- `https://<DSG Management IP address/ESA IP address>/v1/exportGatewayConfigFiles`

This API is introduced in this version. Also, the API uses a stronger cipher and key.

26.2 API for Exporting the CoP Configurations

The API request exports the CoP package that can be used with the DSG containers.

Warning: Do not modify the configuration that has been exported using the CoP API. If you try to modify the exported configuration, then the configuration might get corrupted.

Base URL

`https://<ESA IP address>/exportGatewayConfigFiles`

Method

POST

Curl request syntax

Export API- PKCS5

```
curl --location --request POST 'https://<ESA_IP_address>/exportGatewayConfigFiles' \
--header 'Authorization: Basic <base64 encoded admin credentials>' \
--header 'Content-Type: text/plain' \
--data-raw '{
"nodeGroup": "<node group name>",
"kek": {
"pkcs5": {
"passphrase": "<passphrase>",
"salt": "<salt>"
}
}' -k -o <Response file name in .json format>
```

Authentication credentials

username

Basic authentication user name

Note: In this scenario, the basic authentication is used and so the basic authentication user name is provided. The user can also use the *Client Certificate-based authentication* or *Token-based authentication*.

For more information about supported authentication method for CoP API, refer to the section [Supported Authentication Methods for CoP API](#).

password

Basic authentication password

Note: In this scenario, the basic authentication is used and so the basic authentication password is provided. The user can also use the *Client Certificate-based authentication* or *Token-based authentication*.

For more information about supported authentication method for CoP API, refer to the section [Supported Authentication Methods for CoP API](#).

Request body elements

nodeGroup name and version

Set the node group name and version for which the configurations should be exported.

Note: In the request body element, the user can specify the node group name and version or only the node group name. If the node group name and version parameter is removed from the request body elements, then it will provide the *active* configurations.

Encryption Method

The **pkcs5** encryption is used to protect the exported file.

Note: In this release, the *publicKey* encryption is not supported.

Encryption Method	Sub-elements	Description
pkcs5	salt	Set the salt that will be used to encrypt the exported CoP.
	passphrase	<p>Set the passphrase that will be used to encrypt the exported CoP.</p> <p>Note: The passphrase must be at least ten character long. The passphrase must contain at least one uppercase letter, one numeric value, and a special character.</p>

26.3 API for Exporting the CoP Configurations Using New Version

The new version of the API request exports the CoP package that can be used with the DSG containers.

Warning: Do not modify the configuration that has been exported using the CoP API. If you try to modify the exported configuration, then the configuration might get corrupted.

Base URL

<https://{{ESA IP address}}/v1/exportGatewayConfigFiles>

Method

POST

Curl request syntax

Export API- PKCS5

```
curl --location --request POST 'https://<ESA_IP_address>/v1/exportGatewayConfigFiles' \
--header 'Authorization: Basic <base64 encoded admin credentials>' \
--header 'Content-Type: text/plain' \
--data-raw '{
  "nodeGroup": "<node group name>",
  "kek": {
    "pkcs5": {
      "passphrase": "<passphrase>",
      "salt": "<salt>"
    }
  }
}' -k -o <Response file name in .json format>
```

Authentication credentials



username

Basic authentication user name

Note: In this scenario, the basic authentication is used and so the basic authentication user name is provided. The user can also use the *Client Certificate-based authentication* or *Token-based authentication*.

For more information about supported authentication method for CoP API, refer to the section [Supported Authentication Methods for CoP API](#).

password

Basic authentication password

Note: In this scenario, the basic authentication is used and so the basic authentication password is provided. The user can also use the *Client Certificate-based authentication* or *Token-based authentication*.

For more information about supported authentication method for CoP API, refer to the section [Supported Authentication Methods for CoP API](#).

Request body elements**nodeGroup name and version**

Set the node group name and version for which the configurations should be exported.

Note: In the request body element, the user can specify the node group name and version or only the node group name. If the node group name and version parameter is removed from the request body elements, then it will provide the *active* configurations.

Encryption Method

The **pkcs5** encryption is used to protect the exported file.

Note: In this release, the *publicKey* encryption is not supported.

Encryption Method	Sub-elements	Description
pkcs5	salt	Set the salt that will be used to encrypt the exported CoP.
	passphrase	Set the passphrase that will be used to encrypt the exported CoP. Note: The passphrase must be at least ten character long. The passphrase must contain at least one uppercase letter, one numeric value, and a special character.

26.4 Scenarios for Exporting the CoP Configurations

The CoP package can be exported for the following three scenarios.

Scenario 1: Exporting the *active* CoP configuration

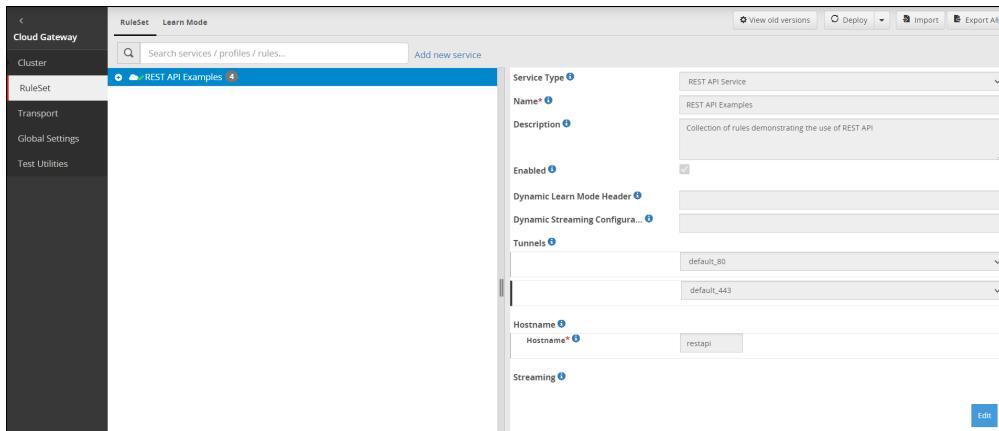


Figure 26-1: The active Ruleset configuration

Sample CURL request

```
curl --location --request POST 'https://xxx.xxx.xxx.xxx/v1/exportGatewayConfigFiles' \
--header 'Authorization: Basic YWRtaW46YWRtaW4xMjM0' \
--header 'Content-Type: text/plain' \
--header 'Cookie: SameSite=Strict' \
--data-raw '{
"kek": {
"pkcs5": {
"passphrase": "xxxxxxxxxxxx",
"salt": "salt"
}
}
'
-k -o cop_demo.json
```

Sample response

```
{
  "cop_version": "1.0.0.0.1",
  "timestamp": 1638782764,
  "cop": {
    "dsg_version": "3.1.0.0.x",
    "copPackage": "<Contents of encrypted CoP package in base64 format>"
  }
}
```

Scenario 2: Exporting the CoP configuration for *lob1* node group

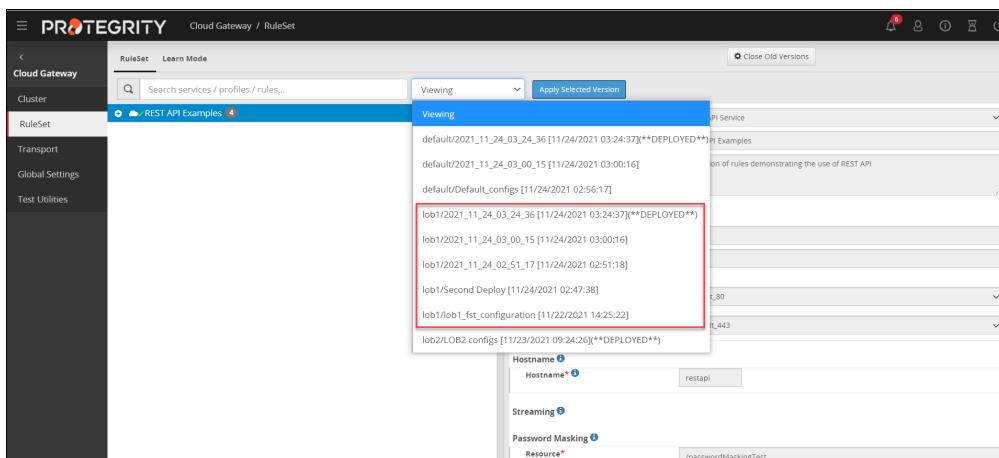


Figure 26-2: The *lob1* configurations

Sample CURL request

```
curl --location --request POST 'https://xxx.xxx.xxx.xxx/v1/exportGatewayConfigFiles' \
--header 'Authorization: Basic YWRtaW46YWRtaW4xMjM0' \
--header 'Content-Type: text/plain' \
--header 'Cookie: SameSite=Strict' \
--data-raw '{
  "nodeGroup": "lob1",
  "kek": {
    "pkcs5": {
      "passphrase": "xxxxxxxxxxxx",
      "salt": "salt"
    }
  }
}'
-k -o cop_demo.json
```

Sample response

```
{
  "nodeGroup": "lob1",
  "cop_version": "1.0.0.0.1",
  "timestamp": 1638796249,
  "cop": {
    "dsg_version": "3.1.0.0.x",
    "copPackage": "<Contents of encrypted CoP package in base64 format>"
  }
}
```

Scenario 3: Exporting the CoP configuration for a particular configuration version of *lob1* node group

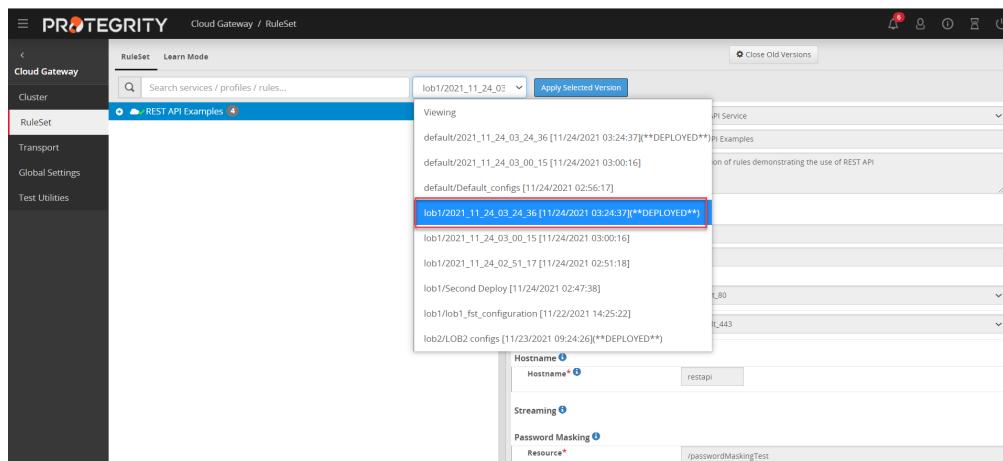


Figure 26-3: The 2021_11_24_03_24_36 configuration version of lob1

Sample CURL request

```
curl --location --request POST 'https://xxx.xxx.xxx.xxx/v1/exportGatewayConfigFiles' \
--header 'Authorization: Basic YWRtaW46YWRtaW4xMjM0' \
--header 'Content-Type: text/plain' \
--header 'Cookie: SameSite=Strict' \
--data-raw '{
  "nodeGroup": "lob1:2021_11_24_03_24_36",
  "kek": {
    "pkcs5": {
      "passphrase": "xxxxxxxxxxxx",
      "salt": "salt"
    }
  }
}'
-k -o cop_demo.json
```

Sample response

```
{  
    "nodeGroup": "lob1",  
    "cop_version": "1.0.0.0.1",  
    "timestamp": 1638796804,  
    "cop": {  
        "dsg_version": "3.1.0.0.x",  
        "copPackage": "<Contents of encrypted CoP package in base64 format>"  
    }  
}
```



Chapter 27

Appendix I: Migrating the UDFs to Python 3

There are codecs for defining the user defined functions used in the DSG. The language that is currently supported for extraction and transformation is Python. In versions prior to the DSG 3.0.0.0, the UDFs were written in the Python 2 version. From DSG 3.0.0.0, the Python version is upgraded from python 2 to python 3. All the UDFs written in python 2 must be migrated to python 3. The UDFs must be made compatible with python 3 to run the codec accurately.

For more information about user defined extraction payload, refer to the section [*User Defined Extraction Payload*](#).

For more information about user defined transformation, refer to the section [*User Defined Transformation*](#).

For more information about user defined functions, refer to the section [*User Defined Functions \(UDF\)*](#).

Perform the following steps to migrate the UDFs to python 3.

1. Export the UDFs from the older DSG versions.
2. Convert the exported UDFs written in python 2 to python 3.

Note: The user can use the python *2to3* standard library to convert the source code to python 3.

The following figures are the sample extraction UDFs in the DSG 2.4.2 and in the DSG 3.0.0.0. In the DSG 2.4.2, the UDFs are written in python 2. In the DSG 3.0.0.0, the UDFs are written in python 3.

Payload*	User Defined Extraction
Programming Language	Python
Source code* 	<pre>""" Example custom extraction implementation. Extracts words from an input string. """ class UserDefinedExtraction(object): def __init__(self, *args): """ Import Python RE module and compile the RE at object creation time. """ import re self.pattern = re.compile("\w+") def extract(self, context): """ Generator implementation. Takes an input string and splits it into words using RE module. Words are yielded one at a time. """ input = context["input"] cursor = 0 output = list() for wordMatch in self.pattern.finditer(input): output.append(input[cursor:wordMatch.start()]) item = { "value": wordMatch.group() } yield item output.append(item["value"]) cursor = wordMatch.end() output.append(input[cursor:]) context["output"] = "".join(output)</pre>

Figure 27-1: User defined extraction payload in the DSG 2.4.2

Payload*	User Defined Extraction
Programming Language	Python
Source code* 	<pre>""" Example custom extraction implementation. Extracts words from an input string. """ class UserDefinedExtraction(object): def __init__(self, *args): """ Import Python RE module and compile the RE at object creation time. """ import re self.pattern = re.compile(b"\w+") def extract(self, context): """ Generator implementation. Takes an input string and splits it into words using RE module. Words are yielded one at a time. """ input = context["input"] cursor = 0 output = list() for wordMatch in self.pattern.finditer(input): output.append(input[cursor:wordMatch.start()]) item = { "value": wordMatch.group() } yield item output.append(item["value"]) cursor = wordMatch.end() output.append(input[cursor:]) context["output"] = b"".join(output)</pre>

Figure 27-2: User defined extraction payload in the DSG 3.0.0.0

The input and output data is in bytes, hence the regex pattern will be bytes.

3. In the DSG 3.0.0.0, import the converted source code in the UDF codec.