Pega Predictive Diagnostic Cloud Security

A Technical Brief

Build for Change®

Table Of Contents

I. Introduction

Pega Predictive Diagnostic Cloud (Pega PDC) analyzes Pega business applications for possible issues, predicts likely points of failure and recommends remediation actions--all delivered via a turn-key SaaS solution hosted on the secure Pega Cloud. Designed specifically to support Pega-based on-premises or cloud-based applications, Pega PDC complements traditional APM tools to provide a more holistic view of application health.

This brief examines and addresses the principal security concerns related to the monitoring of a Pega application by PDC. In summary:

1) What data is sent to Pega PDC?

- Only application metadata is sent
- Provided that the application has been implemented within Pega Guardrails, no user data is sent to Pega PDC
- The data transmission protocol is strictly one way—from the target application to Pega PDC. Pega PDC does not reach back into source systems for additional data

2) How is data secured during transmission?

- All data is sent to Pega PDC using SOAP over HTTPS, so all communications are fully encrypted
- Browser logins to the system are also done over HTTPS and thus are fully encrypted
- A single point of transmission from within the customer's outer firewall is recommended

3) How is data in Pega PDC secured?

- The Pega PDC system itself is a PRPC application built on Pega 7 and securely hosted in the Pega Cloud®
- Customers use a unique URL and auto-generated alphanumeric passwords to access the system
- Customers only have access to their system data and can never see information or data about other customers

II. Data Sent to Pega PDC

In terms of the data that is sent, Pega does not send user data to the cloud. Pega PDC bases its analysis on several sources of data from monitored Pega systems to provide detailed information about both performance and functional problems that have occurred. Summary performance statistics are also sent regularly to give context about the overall performance of monitored systems. The majority of the data sent consists of counts and times, providing detail for analysis. Pega PDC uses a white list model for sending other important information. This means that Pega only sends parameters that we have previously identified as both necessary for analysis and also with known, safe contents. By taking steps to send the minimum data set needed, Pega reduces the risk of sending sensitive data to Pega PDC. Below we review the specific types of data that is sent to Pega PDC.

A. Alerts

The bulk of data sent to Pega PDC is from Pega Alerts. Pega Alerts are written to the Pega Alert Log and also sent in real time from monitored systems to Pega PDC. There are over 50 types of alerts,

triggered when particular counts or elapsed times exceed a threshold during an interaction. For example, if the server time in an interaction is over one second, an alert will be generated with the relevant statistics so Pega PDC can understand what happened.

Alerts contain metadata about what happened in that interaction. Session context information such as requestor id, application and first activity executed are included. In addition, performance statistics such as elapsed time, cpu time, database time, and count of rules executed are included to measure the impact of the issue as well as help debug what the root cause of the issue is.

Individual alerts are kept in the system for 14 days before being purged. Pega PDC creates work objects for related alerts containing recommendations and historical data. Those items are persisted in the system beyond the original 14 days to provide trend analysis for that particular issue.

You can see the complete data set that is published by your system by viewing the Pega Alert Log for that system. It is recommended that customers inspect the alert log to get familiar with what data is being sent in the alert. Also see the appendix below for an example alert including the exact data being sent to Pega PDC. Refer to https://pdn.pega.com/performance/understanding-alert-log-message-data for complete documentation of the fields sent in alerts.

B. Parameter Page

In addition some parameters from the current parameter page are sent along with the alert. The parameter page contains important contextual information about what functions are being run in the monitored system. For example, if an alert is generated from a slow running report, the name of the report record is pulled from the parameter page and sent to the Diagnostic Cloud. Without knowing the name of the report, all slow reports would be grouped together.

The necessary parameters are sent according to a white list in PRPC. You can configure the white list to send additional parameters as necessary to provide more context around your alerts. The white list can be accessed by looking at the following Data-Admin-System-Setting: prconfig/alerts/parameterpage/allowed .

C. Database Alerts

For database alerts, the database query is sent as part of the alert. Pega uses bind variables in the construction of queries. However, these bind variables are not transmitted to Pega PDC, thus reducing exposure.

D. Node Startup Alert

Each time an application server is started, PRPC sends a node startup alert (Pega0008) to Pega PDC. In addition to the usual alert fields, the server startup alert includes the server name, the system name, the server description, and the SOAP URL for the node. This startup message allows Pega PDC to automatically recognize the different nodes, and categorize them by cluster. This allows Pega PDC users the ability to see alerts and performance statistics broken down by server, so that any irregular patterns can be identified. See the appendix for a sample node startup alert.

E. Exceptions

In addition to alerts, exceptions are also sent to Pega PDC for analysis. DEBUG or INFO statements are not sent, only exceptions. This exception data can be viewed by looking at the "ERROR" lines in the PegaRULES log. PRPC also includes some contextual information such as the requestor id. The contextual data sent in exceptions is a subset of the fields sent for alerts, listed above. See the Appendix for sample Exceptions. Also see the appendix below for a sample exception including the exact data being sent to Pega PDC.

F. Performance Statistics

Summary performance statistics are sent hourly from monitored systems to Pega PDC. These statistics are used to identify overall performance and performance trends of the systems, including statistics such as average response time and unique user count.

You can view the data being sent by running the report "PushLogUsageData", or by inspecting the pr_perf_stats table in the database. An example is also included in the appendix.

G. Database Indices

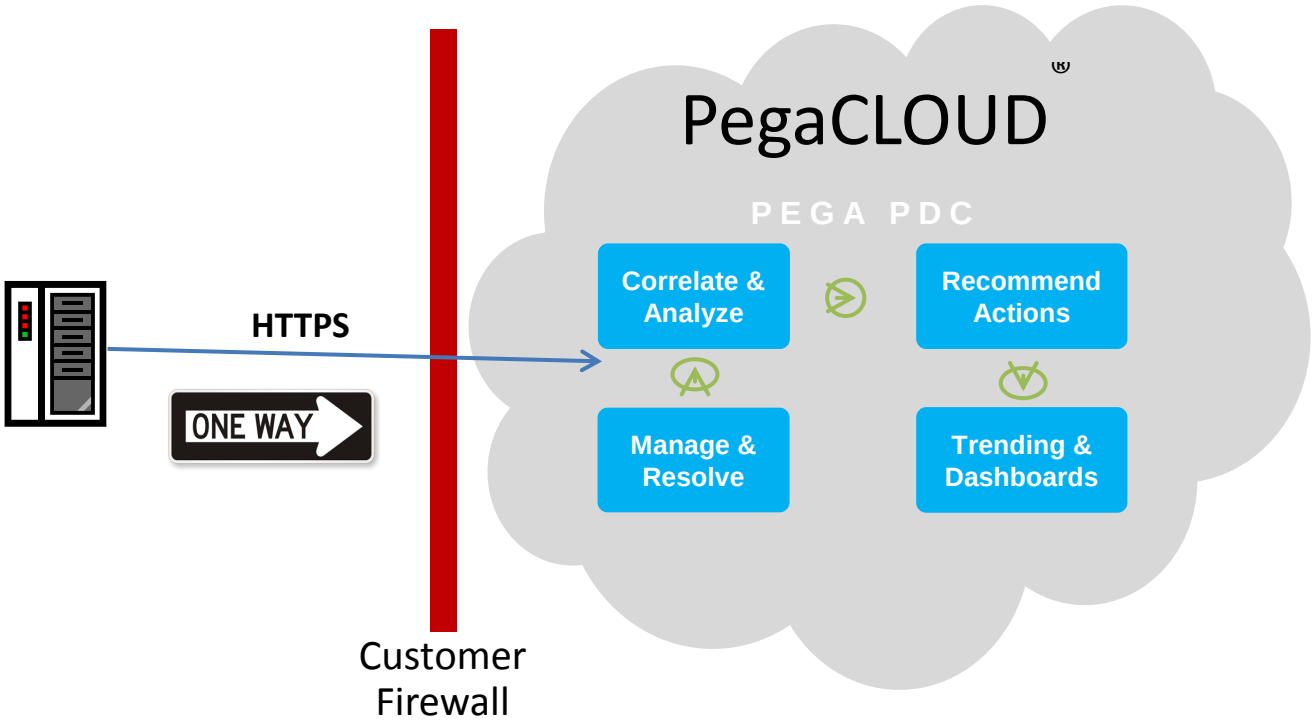
Database index information is gathered daily to help create recommendations for improving query performance. This allows Pega PDC users to see what indexes are currently in use so they can identify if other indexes are necessary. By sending the database indexes, users do not have to get separate credentials to access the database, or go through another resource to get the index details. The activity "PushDBIndexes" gathers the index information for all tables, and sends it to Pega PDC. An example of the data being sent can be found in the appendix of this document. Also see the appendix below for an example of a database index message sent to Pega PDC.

H. Guardrail Counts

In Pega 7, we also push guardrail warning counts to Pega PDC. These guardrail warnings point to Pega best practices that are not being followed in rules being developed. These counts include 7 numbers: the total rule count as well as the separate counts of the rules with justified and unjustified warnings of severe, moderate, and caution. See the appendix below for a sample guardrail message sent to Pega PDC.

III. Data Transmission Methods

Data is sent to Pega PDC using SOAP over HTTPS, so all communications are fully encrypted. This will require allowing SOAP messages to be sent from your monitored PRPC systems to Pega PDC. Information flow for Pega PDC is one-way; Soap messages are sent from source systems to Pega PDC, but Pega PDC does not reach back into the source system for additional information. Users interact with Pega PDC directly by logging in through their web browser over an encrypted https connection, or subscribing to emailed reports.



IV. Security of the Pega PDC System

The Pega PDC system itself is a PRPC application built on Pega 7 and securely hosted in the Pega Cloud® on servers in the United States. When customers sign up for Pega PDC, they are given a unique URL to log in to the system. Customers are segregated and only have access to their system data based on their unique URL and can never see information or data about other customers. Unique alphanumeric passwords are also auto-generated and sent via a separate email.

There are two portals available in Pega PDC, manager and user portals, which allow for granular access management. From the manager portal operators can be added and deleted. Operator passwords can also be reset from this portal. Managers can restrict access for users to specific systems. For example, you may have different applications on different Pega systems connected to Pega PDC. In that case you may want developers for system A to only be able to view data for system A, and not for system B. This configuration can be done in the "Manage Operators" menu in the manager portal.

V. Related Links

Pega PDC Forum:

https://pdn.pega.com/forums/pega-predictive-diagnostic-cloud

VI. Appendix

A. Fields Sent in Alerts

Field	Description	Alert Version	Example
Date Timestamp (GMT)	The time that the alert was generated in the system, translated into GMT from the system time.	1	2007-11-06 06:11:55,436 GMT
Version	The version of the header for the alert. This information is used to tell the system which "parse" rule to use to parse this message (in case different information is added into the messages in the future)	1	5
Message ID	The message key for the type of alert.	1	PEGA0004
KPI Value	The value of the Key Performance Indicator (KPI), which triggered the alert.	1	50021042
KPI Threshold	The threshold value of the "Key Performance Indicator", as set in the prconfig.xml file.	1	50000000
System Node ID	The node (hash) ID in the Process Commander system.	1	ed3d462358e52d133 c4d805bcaf1dc40
Requestor ID	The hash name of the requestor. If the name begins with "H", this requestor is being used by a user (HTTP interaction) If the name begins with "A", this requestor is being used by an application (example: SOAP service) If the name begins with "B" this is a batch requestor (used by agent processing) If the name begins with "P", this requestor is used for portlet support.	1	HA39AD8D65146D5 8B903281E806DD9DCD
User Name	The user ID associated with the requestor for which the alert occurred. (Only if	1	theDeveloper@pegasystems.com

Field	Description	Alert Version	Example
	available. Some alerts may not have a userID)		
Work Pool	Work Pool the user was in at the time of the alert generation if available.	4	PegaSample
Rule Application	Rule Application used at time of alert.	6	PegaRULES:07.10
Encoded Ruleset List	The encoded rule set list of the user.	4	bd72de7fe35adfdfcb28b 7486a6b7a20
Allows Rule Check Out	Flag that identifies if the user is allowed to check out rules.	4	N
Interaction Number	The interaction number for this interaction with the server (can be used to match up this data with PAL readings or DBTrace).	2	30
Unique Alert Number	In order to ensure that a unique number is generated to identify each alert in a heavily-used system, there are three parts to the key definition of each alert: • date/timestamp • node name of the system • unique alert number Adding the unique number to the other two fields guarantees individuality.	1	185
Thread Name	The name of the thread on which the alert happened.	1	http-80-31
Pega Thread Name	The name of the Pega thread on which the alert happened.	6	STANDARD
Logger Name	Usually, the path to a Java class file (but doesn't have to be). This shows the java class inside which the alert was generated.	1	com.pega.pegarules.engine. database.DatabaseImpl
Stack	An indicator in the engine process showing the state of processing when the	1	10.1.1.1|10.1.1.11

Field	Description	Alert Version	Example
	message occurred. (May not be available for all alerts.)		
Last Input	Stream or activity that is referenced in the URL that started the interaction.	4	Activity=FinishAssignment
First Activity	The first activity that was executed.	4	Rule-Obj-Activity:FinishAssignment
Last Step	The last step that was executed.	4	EMBED-LISTPARAMS GETCONTENT #20070809T151219.534 GMT Step: 1 Circum: 0
Trace List	An internal trace of the most recent actions executed by the interaction.	4	441:haveAccessWithFrame :Rule-HTML-Fragment;442:getStream Rule-HTML-Section:ShowListView;
PAL	A snapshot of PAL at the time of the alert.	4	RuleCPU=0.03;RDBWithStreamCount=1 ;RuleFromCacheCount=32;
Primary Page Class	Class of the primary page at the time of the alert.	5	Embed-ListParams
Primary Page Name	Name of the primary page at the time of the alert.	5	ListParamsPage
Step Page Class	Class of the step page at the time of the alert.	5	Embed-ListParams
Step Page Name	Name of the step page at the time of the alert.	5	ListParamsPage
Pega Stack Trace	The pega stack at the time of the alert.	5	java;RULE-OBJ-ACTIVITY EMBED-LISTPARAMS GETCONTENT #20070809T151219.534 GMT Step: 1 Circum: 0;
Variable text	Information specific to this alert message is stored in a variable length string.	1	The number of database bytes input for this interaction has exceeded the "warning" level for Requestor HA39AD8D65146D58 B903281E806DD9DCD, operator developer@pega.com Maximum bytes: 50000000 Actual bytes: 50021042

B. Sample Node Startup Alert

2013-10-10 04:44:00,842 GMT*7*PEGA0008*0*0*368aa03b0c370380e19aba5ae40e

ea68*NA*NA*B0C31C7C1C2B4B356D7E4CBF7FF7C8AA1*n
one*PegaSample*null*4ff0c05daaa08916c2fe59eb7b39aaa
c*N*0*23*appServer*STANDARD*com.pega.pegarules.sess
ion.internal.engineinterface.etier.impl.EngineStartup*NA*(
License
Daemon)****NA*NA*NA*NA*NA*NA*NA*PegaRULES
engine successfully started. Server: appServer System: pega
Description: appServer pega 2013-10-01 15:29:34.256 GMT
URL: http://localhost:8080/prweb/PRSOAPServlet*

C. Sample Exception

2014-07-01 17:48:44,001 GMT*2*ae39681a112234d1234b12345678e12*http-apr-8080-exec-3*com.pega.pegarules.session.internal.engineinterface.service.HttpAPI*H50B123304321AB131876C23D4F2A2345**STANDARD*PegaRULES:07.10*Administrator@pega.com*10.3.1.1|10.1.1.1*10.1.1.1: com.pega.pegarules.pub.PRRuntimeError

com.pega.pegarules.pub.PRRuntimeError: PRRuntimeError

at com.pega.pegarules.session.internal.mgmt.base.ThreadRunner.runActivitiesAlt (ThreadRunner.java:706)

at com.pega.pegarules.session.internal.mgmt.PRThreadImpl.runActivitiesAlt (PRThreadImpl.java:433)

D. Sample Performance Statistics Message

<LogUsageView>

<NodeName>Node1-1</NodeName>

<NodeID>ab12345e694176d1763b611099320e84</NodeID>

<Date>6/9/14 4:16:43 PM EDT</Date>

<Entries>

<Entry Day="20140609" pxRequestorType="APP" pxSystemNodeIDMax="ab12345e694176d1763b611099320e84" pxSystemNodeMax="Node1-1" pxSystemNameMax="pega" pxProcessCPUMax="0" UniqueRequestors="2" UniqueUsers="0">

<NetSegment pxActivityCount="0" pxAlertCount="0" pxCommitCount="0" pxCommitElapsed="0" pxConnectCount="0" pxConnectElapsed="0" pxDBInputBytes="0" pxDBOutputBytes="0" pxDeclarativeRulesInvokedCPU="0" pxFlowCount="0" pxInputBytes="0" pxInteractions="0" pxJavaAssembleCount="0" pxJavaCompileCount="0" pxOtherBrowseElapsed="0" pxOtherBrowseReturned="0" pxRuleCount="0" pxRuleIOElapsed="0" pxServiceCount="0" pxTotalReqCPU="0" pxTotalReqTime="0"

pxOtherIOCount="0" pxOtherIOElapsed="0" pxOutputBytes="0" pxRuleBrowseElapsed="0" pxRuleBrowseReturned="0"/>

</Entry>

<Entry Day="20140609" pxRequestorType="BATCH" pxSystemNodeIDMax="ab12345e694176d1763b611099320e84" pxSystemNodeMax="Node1-1" pxSystemNameMax="pega" pxProcessCPUMax="0" UniqueRequestors="44" UniqueUsers="1">

<NetSegment pxActivityCount="1255" pxAlertCount="3" pxCommitCount="10" pxCommitElapsed="0.129217" pxConnectCount="0" pxConnectElapsed="0" pxDBInputBytes="228313863" pxDBOutputBytes="553055" pxDeclarativeRulesInvokedCPU="2" pxFlowCount="0" pxInputBytes="0" pxInteractions="1482" pxJavaAssembleCount="53" pxJavaCompileCount="53" pxOtherBrowseElapsed="0.252447" pxOtherBrowseReturned="4" pxRuleCount="942" pxRuleIOElapsed="0.167707" pxServiceCount="0" pxTotalReqCPU="6.536441" pxTotalReqTime="8.683186" pxOtherIOCount="1543" pxOtherIOElapsed="2.382814" pxOutputBytes="0" pxRuleBrowseElapsed="6.471477" pxRuleBrowseReturned="7911"/>

</Entry>

<Entry Day="20140609" pxRequestorType="BROWSER" pxSystemNodeIDMax="ab12345e694176d1763b611099320e84" pxSystemNodeMax="Node1-1" pxSystemNameMax="pega" pxProcessCPUMax="0" UniqueRequestors="3" UniqueUsers="1">

<NetSegment pxActivityCount="2327" pxAlertCount="14" pxCommitCount="8" pxCommitElapsed="0.075725" pxConnectCount="1" pxConnectElapsed="1.147813" pxDBInputBytes="88421174" pxDBOutputBytes="2375979" pxDeclarativeRulesInvokedCPU="420" pxFlowCount="4" pxInputBytes="154255" pxInteractions="151" pxJavaAssembleCount="986" pxJavaCompileCount="993" pxOtherBrowseElapsed="0.019899" pxOtherBrowseReturned="4" pxRuleCount="7220" pxRuleIOElapsed="1.84892" pxServiceCount="0" pxTotalReqCPU="547.688311" pxTotalReqTime="556.055227" pxOtherIOCount="4505" pxOtherIOElapsed="5.656485" pxOutputBytes="5474466" pxRuleBrowseElapsed="0.221692" pxRuleBrowseReturned="10"/>

</Entry>

</Entries>

</LogUsageView>

E. Sample Database Index Message

<IndexInfo>
<ClusterName>pega</ClusterName>
<Entries>

<Entry TableName="PR_SYS_APP_HIERARCHY_FLAT"
IndexName="PR_SYS_APP_HIERARCHY_FLAT_IDX2" ColumnName="PZAPPHASH">
</Entry>
<Entry TableName="PR_SYS_APP_HIERARCHY_FLAT"
IndexName="PR_SYS_APP_HIERARCHY_FLAT_IDX3" ColumnName="PZBUILTONAPPHASH">
</Entry>
[Truncated for length]
</Entries>
</IndexInfo>

F. Sample Guardrail Message

<GuardrailView>
<Date>6/10/14 1:00:00 AM EDT</Date>
<NodeID>ae31234e694176d1763b611099320e84</NodeID>
<TotalRuleCount>2444</TotalRuleCount>
<Entries>
<Entry IsJustified="false" WarningCount="5"
WarningSeverity="2"/>
<Entry IsJustified="false" WarningCount="5"
WarningSeverity="3"/>
</Entries>
</GuardrailView>