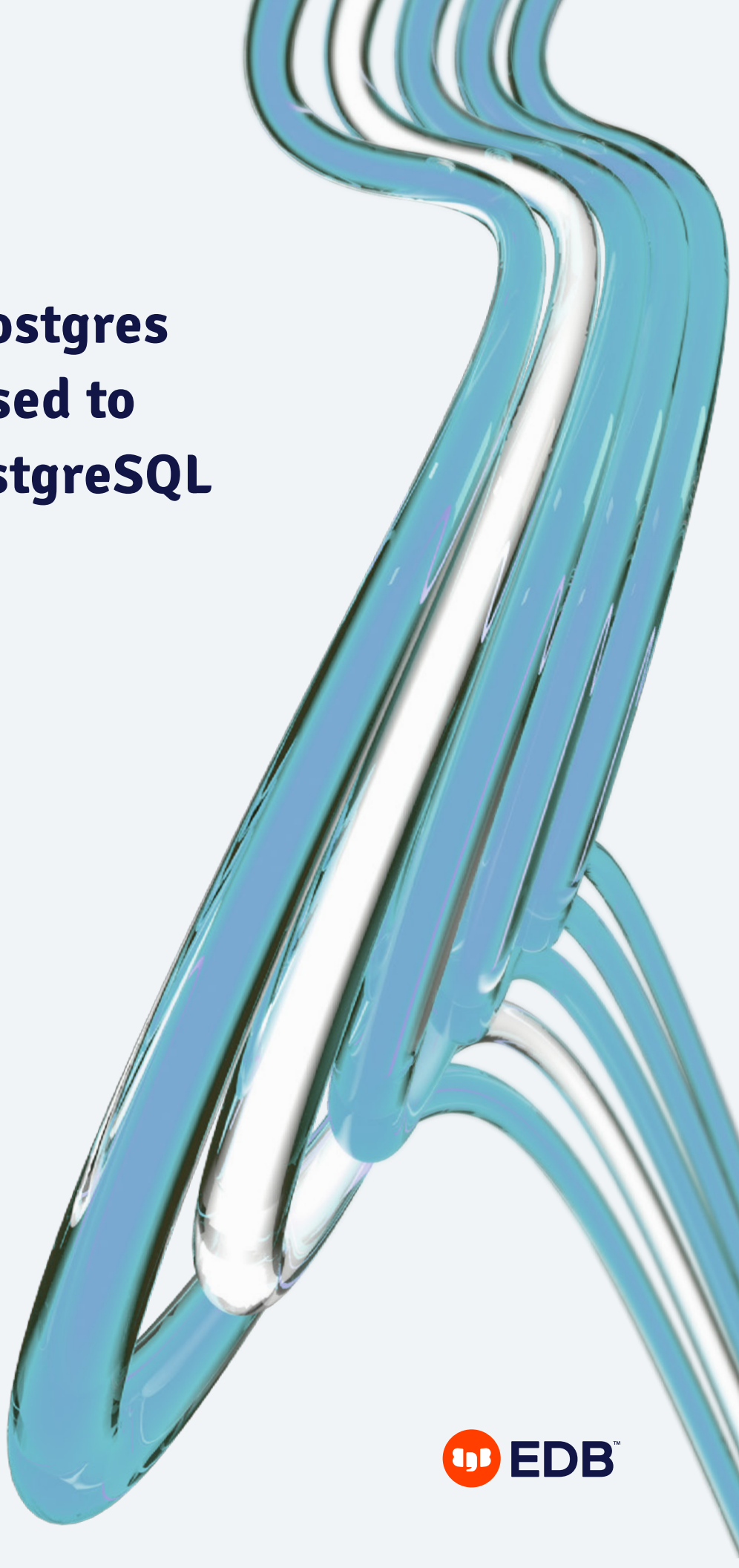
How the EDB Postgres STIGs Can be Used to Secure Your PostgreSQL Database

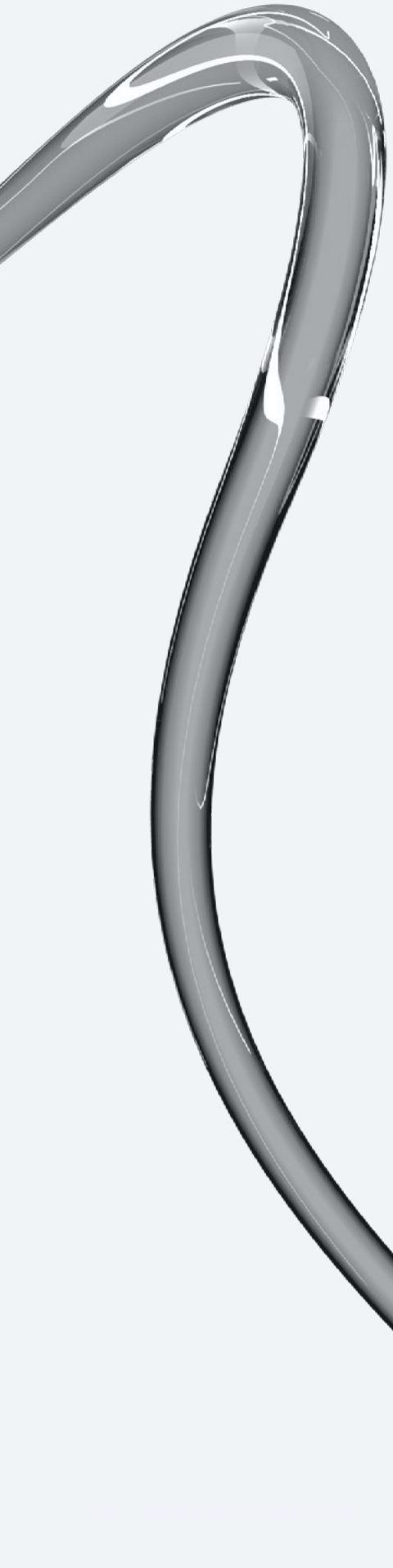


AUTHORED BY:

Matthew Lewandowski
Field CTO, EDB

POWER TO POSTGRES

EDB™



Contents

1 Introduction

Increasingly organizations are turning to the use of PostgreSQL for their database needs. As my colleague, Simon Riggs here at EDB says, "PostgreSQL's speed, security and robustness make it suitable for 99% of applications, so it's a great starting place for any application." This suitability is one of the main reasons why PostgreSQL has become so popular. PostgreSQL's inherent security features are why organizations are able to use it with confidence.

Although a particular database management system (DBMS) may include a rich set of security features, it is still up to the organizations using the DBMS to take advantage of those features to secure their deployments of it. EDB has published several blogs and other materials that provide some high level guidance on securing Postgres using the security features provided in the open source version of PostgreSQL as well as those exclusively available in the EDB Postgres Advanced Server version. I highly recommend reviewing the excellent How to Secure PostgreSQL: Security Hardening Best Practices & Tips blog post written by my colleague Dave Page, and the Security Best Practices for PostgreSQL white paper.

2 Why Should You Care About STIGs?

To assist United States Department of Defense (DoD) organizations with implementing secure deployments of products used in their systems, the Defense Information Systems Agency (DISA) has created a set of Security Requirements Guides (SRGs) and related Security Technical Implementation Guides (STIGs) for different technology areas, including databases. All products used within DoD systems are required to comply with the requirements specified in the SRGs.

While these guides have been created specifically for the U.S. DoD, many other non-DoD organizations in the public and private sectors also refer to them for guidance on securing their systems.

3 What Are The EDB Postgres STIGs?

The first paragraph in section 1.1.1 of the Database SRG (V3R1) Overview document provides a good summary description of the SRGs and STIGs and how they relate to other security publications:

"SRGs are collections of requirements applicable to a given technology family. SRGs represent an intermediate step between

Control Correlation Identifiers (CCIs) and Security Technical Implementation Guides (STIGs). CCIs represent discrete, measurable, and actionable items sourced from Information Assurance (IA) controls defined in a policy, such as the National Institute of Standards and Technology (NIST) Special Publication (SP) 800-53. STIGs provide product-specific information for validating and attaining compliance with requirements defined in the SRG for that product's technology area."

For database management systems, DISA created the Database SRG as part of the Application family of SRGs. The Database SRG contains descriptions of vulnerabilities applicable to databases and general check and fix statements that can be used to address each of the vulnerabilities for database products that do not have a STIG. Product vendors that have an identified sponsoring DoD organization work with DISA to create and approve a STIG that contains product-specific checks and fixes.

In July of 2016, EDB obtained approval for an EDB Postgres Advanced Server (EPAS) STIG, which at the time was the first STIG approved for a version of Postgres. The following year, DISA approved a STIG for open source PostgreSQL that was developed with DISA by Crunchy Data and Pivotal Software. Many of the checks and fixes identified in the PostgreSQL STIG are similar to those in the EPAS STIG; however, the EPAS STIG includes checks and fixes that are able to leverage additional features available with the EDB platform, especially those related to security.

Both the original EPAS STIG and the PostgreSQL STIG provide checks and fixes specific to deployments on Red Hat Enterprise Linux (RHEL) systems. To address the needs of customers who are deploying EPAS on Windows, EDB worked with DISA to develop and approve an EPAS STIG for Windows. The EPAS STIG on Windows was approved in May of last year (2020).

Although the EPAS STIG for Windows does include check and fix statements that are EPAS specific, many of the checks and fixes are also applicable to open source PostgreSQL as well. As such it can be used as a starting point for configuring a PostgreSQL deployment on Windows. There are some vulnerabilities that would need to be addressed in other ways. There are some vulnerabilities that would need to be addressed, with some being harder than others. For example, EPAS is the only version of Postgres on Windows that provides FIPS 140-2 validated encryption modules. Several of the vulnerabilities in the database SRG require the use of FIPS 140-2 encryption.

4 How Do You Work with a STIG?

Downloading and Viewing

In order to work with a STIG, you need to first obtain a copy of it or access an online version of it. There are two main options available. They are described in the following sections.

Option 1 - DoD Cyber Exchange Site

The official STIG document library site that is accessible to the public is on the DoD Cyber Exchange site at https://public.cyber.mil/stigs/downloads/. This page contains a searchable list of the available STIGs and related documents with hyperlinks for downloading. The download links on this site point to zip files that contain the actual STIG in an XML formatted file. The following figure shows an example of the contents of the zip file when it is expanded.

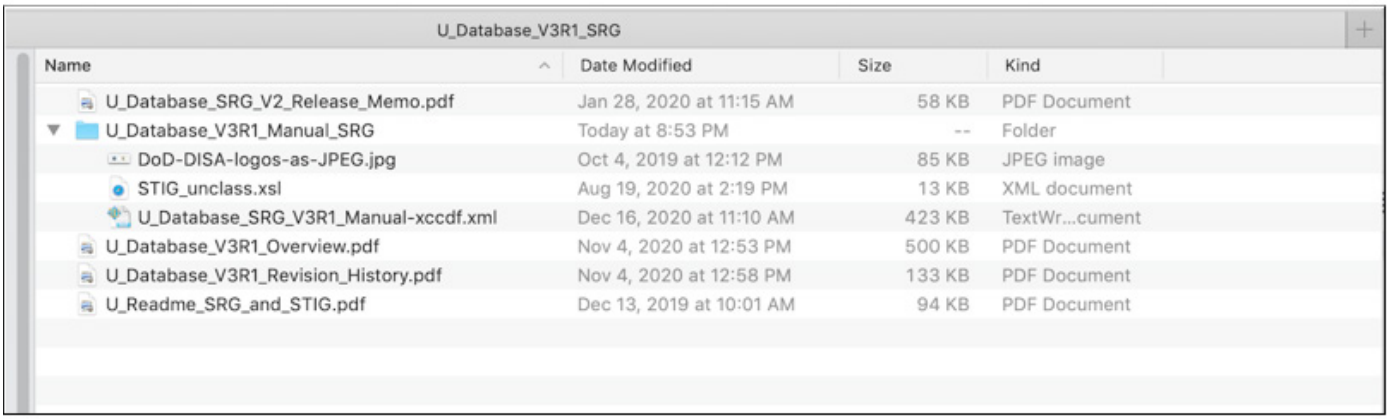


Figure 1 - Example Contents of an SRG or STIG Package Downloaded from the DoD Cyber Exchange STIG Document Library

A STIG zip file from the DoD Cyber Exchange site typically includes the following content:

- PDF formatted Release Memo
- PDF formatted Overview document
- PDF formatted Revision History document
- A *Manual* Folder
- XML formatted STIG document
- XSL file used to transform the XML file for display in HTML
- JPEG logo images referenced by XSLT
- Supplementary materials and documentation if available
- Possibly other documents

The STIG XML file format is in Extensible Configuration Checklist Description Format (XCCDF), a NIST led specification development project. According to the description provided on the NIST XCCDF page, the XCCDF specification is "designed to support information interchange, document generation, organizational and situational tailoring, automated compliance testing, and compliance scoring." The XCCDF is intended to support ingestion into Security Content Automation Protocol (SCAP) tools for automated compliance validation purposes.

The SRG/STIG Tools page on the DoD Cyber Exchange site provides some information on how to open and view the contents of the XML file in a more human readable format. This page also includes links for downloading a Java based STIG Viewer tool. This tool can be used to import an SRG or STIG XML file into it for viewing and searching. It can also be used to export the selected SRG or STIG to XML, RTF, or HTML formatted files. If you prefer not to download and use the STIG Viewer tool, it is also possible to open the XML file and have it displayed as HTML in a web browser like Firefox.

Note that more recent versions of Firefox may require toggling the privacy.file_unique_origin advanced preference (see: Mozilla Firefox support page for "Why is my XSL file no longer being applied to my XML file?" issue).

Option 2 - UCF STIG Viewer Site

As an alternative to downloading an SRG or STIG from the DoD Cyber Exchange site, the Unified Compliance Framework (UCF) provides an online STIG Viewer site. This site lists the available SRGs and STIGs as hyperlinks for viewing the content of the selected STIG via the browser. Selecting a STIG displays an overview of the STIG as well as a list of each finding (i.e., vulnerability) in the STIG for the selected Mission Assurance Category (MAC) profile with hyperlinks for viewing the details of the selected vulnerability. The overview section on the page where the findings are listed also includes hyperlinks for downloading the STIG contents as Excel (actually CSV), JSON, or XML formatted files.

5 Anatomy of a STIG and a STIG Rule

To better understand how to use a STIG, it is important to have a basic understanding of its content. An SRG or a STIG basically contains a set of rules that must be complied with. Each rule is meant to address a potential vulnerability. If a system fails to comply with a particular rule for a vulnerability, this is considered a finding. Although they have slightly different meanings, the terms “Vulnerability,” “Rule,” and “Finding” are often used to refer to a single record in an SRG or STIG. The following figure provides an example of a STIG vulnerability found in the EDB Postgres Advanced Server STIG for Windows.

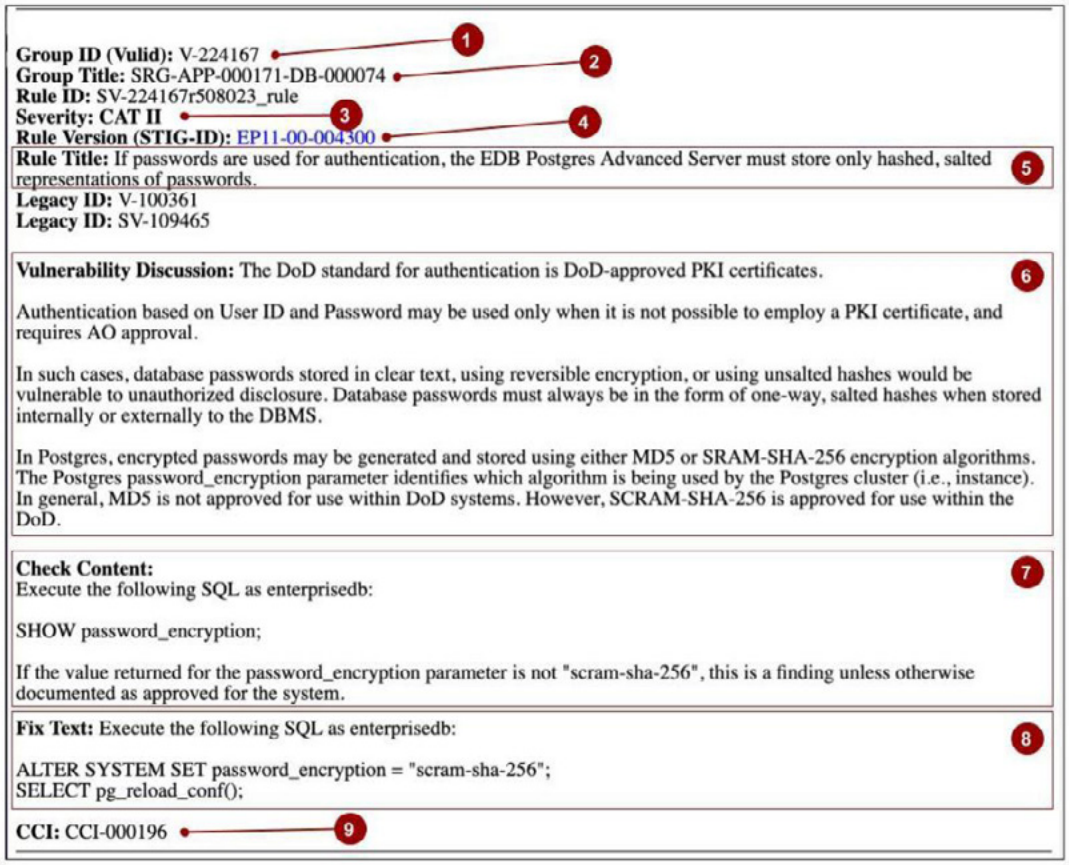


Figure 2 - Example STIG Vulnerability Rule from the EDB Postgres Advanced Server STIG for Windows

The fields labeled with a number in the figure above are described on the following pages.

1. **Group Id (Vulid)**

   The Group Id is a unique identifier for the vulnerability in the system used by DISA to produce the STIGs. Note that in some systems that can be used to view the contents of a STIG, this field may be referred to as a "Finding Id."

2. **Group Title**

   The Group Title is an alphanumeric identifier for the rule identifying the corresponding parent (or group) rule version for the particular SRG or STIG rule. The naming format of the Group Title makes it easy to identify the SRG hierarchy corresponding to the vulnerability. For example, in the V3R1 Database SRG, the record with Group Id of "V-224167" (Rule Version: SRG-APP-000171-DB-000074) has a Group Title of "SRG-APP-000171," which corresponds to a rule in the top level Application Security Guide with a Rule Version of "SRG-APP-000171" (Group Id: V-26923). The EDB Advanced Server STIGs then have rules with a group title of "SRG-APP-000171-DB-000074" that correspond to that rule in the Database SRG. When comparing check content and fix text across different versions of STIGs for a particular product category like databases (say, between EPAS on RHEL and EPAS on Windows or EPAS on RHEL and PostgreSQL), the Group Title is a common identifier that can be used to identify the same SRG rule.

3. **Severity**

   Each vulnerability rule is assigned a severity category code. The following table describes the severity values, their category code, and related DISA guidelines for the category code (from Database SRG Overview document section 1.3).

Severity Value	Severity Category Code	DISA Category Code Guidelines
high	**CAT I**	Any vulnerability, the exploitation of which will **directly and immediately** result in loss of Confidentiality, Availability, or Integrity.
medium	**CAT II**	Any vulnerability, the exploitation of which **has a potential** to result in loss of Confidentiality, Availability, or Integrity.
low	**CAT III**	Any vulnerability, the existence of which **degrades measures** to protect against loss of Confidentiality, Availability, or Integrity.

When applying the STIGs, all CAT I and CAT II findings need to be resolved.

The following table identifies the number of rules in each severity code category in the latest current versions of the Database SRG, both EPAS STIGs, and the community PostgreSQL STIG. If you look at the numbers in the table, you'll notice that the total number of rules for each publication is different. The reason for this will be explained later.

SRG / STIG	# of Rules		
	CAT I	CAT II	CAT III
Database SRG (V3R1)	0	124	0
EPAS on RHEL STIG (V2R1)	9	99	0
EPAS on Windows STIG (V2R1)	5	76	0
PostgreSQL (V2R1)	7	103	0

4. **Rule Version**

   The Rule Version, which is also known as the STIG-ID, is an identifier specified with the same basic format as the Group Id used to identify the SRG or STIG specific rule. For example, in the Database SRG, there is a rule with a Rule Version value of "SRG-APP-000171-DB-000074" and in the EPAS STIG for RHEL and EPAS STIG for Windows there corresponding rules with Rule Version values of "PPS9-00-004300" and "EP11-00-004300," respectively.

5. **Rule Title**

   The Rule Title is usually a one-sentence summary of the rule. The Rule Title in the STIGs will identify the specific product (e.g., EDB Postgres Advanced Server) as opposed to the more generic reference of "DBMS" specified in the SRG Rule Title.

6. Vulnerability Discussion

The Vulnerability Discussion section contains background information about the vulnerability and the intent of the rule. In the SRG, the discussion is of a general nature. Oftentimes, the text in the STIG is simply a copy of the text in the SRG with the reference of "DBMS" replaced by the specific product for which the STIG applies. However, in some cases, vendors provide additional product specific information to help provide clarity or rationale behind the checks and fixes that have been identified for the vulnerability.

7. Check Content

The Check Content section identifies the steps to take to determine whether the system is in compliance with the rule. Oftentimes, the check will involve running one or more OS commands or database commands to report or view OS level or database level settings (e.g., configuration, permissions, etc.). If available, GUI based tools may be used to obtain the results as well. The check content may also involve verifying that the results returned by a particular command or set of commands are documented and approved in the system security documentation or other system documentation. In some cases, the check may require inspecting an application's code and design to verify that interactions with the database comply with the rule. If a check reveals that the system is not in compliance with the rule, this is considered as a finding.

8. Fix Text

The Fix Text section identifies the steps to take to correct a finding that the system is not in compliance with the rule. In most cases, the fix will involve running commands or using GUI based tools to update the database or OS level settings to be in compliance with the rule. In some cases, the fix may involve changes to the application or operating procedures to meet the requirement. Additionally or alternatively, in some cases, the system documentation may need to be updated to indicate that the current or updated setting is approved along with the rationale and justification for the setting. Some of the identified fixes will be very explicit about the setting or configuration that is required. Other fixes may depend on organizational policies and application requirements.

9. CCIs

The CCI section lists the DISA identified Control Correlation Identifiers (CCIs) that correspond to the particular vulnerability rule. CCIs provide a mechanism for mapping the rules in the SRGs and related STIGs to higher level Information Assurance controls. In the case of the Database SRG and STIGs, the rules map to controls in the NIST Special Publication (SP) 800-53, Revision 4. The following table provides a breakdown of the number of rules in the Database SRG, the EPAS STIGs, and the community PostgreSQL STIG by the control families identified in the NIST SP 800-53 publication.

Control Family Id	Control Family Name	# of Rules			
		Database SRG (V3R1)	EPAS on RHEL STIG (V2R1)	EPAS on Windows STIG (V2R1)	PostgreSQL STIG (V2R1)
AC	Access Control	11	10	11	10
AU	Audit and Accountability	57	54	26	54
CM	Configuration Management	15	15	17	14
IA	Identification and Authentication	12	10	12	10
SC	System and Communications Protection	20	12	13	15
SI	System and Information Integrity	9	7	7	8

As can be seen in the table, a significant portion of the rules are related to auditing.

6 Applying the STIG

When deploying a database as part of their applications or systems, U.S. DoD organizations need to comply with each of the rules in the database SRG. Non DoD organizations may also desire to comply with these rules since they serve to identify a generic set of security best practices. The STIGs are helpful in that they provide product specific checks and fixes for complying with these rules. Note, however, that a STIG only includes rules that have a status of "Applicable - Configurable." This status means that it is possible to configure the product and related components in order to be compliant with the rule. If an SRG rule does not have a corresponding STIG rule in a given product STIG, this means that the rule is either inherently met by the product (i.e., no additional configuration required) or that the rule cannot be met by the product. Non-product provided solutions will need to be employed to meet the SRG rules that are not inherently met or cannot be met through the configuration of the product.

The basic process of applying the STIG, which is depicted below, is to go through each rule in the STIG and perform the checks that are prescribed in the Check Content section of the rule. Then the corrective actions identified in the Fix Text for a rule are performed to address any findings (i.e., system not in compliance) that may have been identified by the checks. After applying the STIG specific checks and fixes, the generic checks and fixes identified in the SRG should be consulted for guidance on addressing the potential vulnerabilities identified for those rules that do not have a corresponding set of checks and fixes in the STIG and are not inherently met by the product.

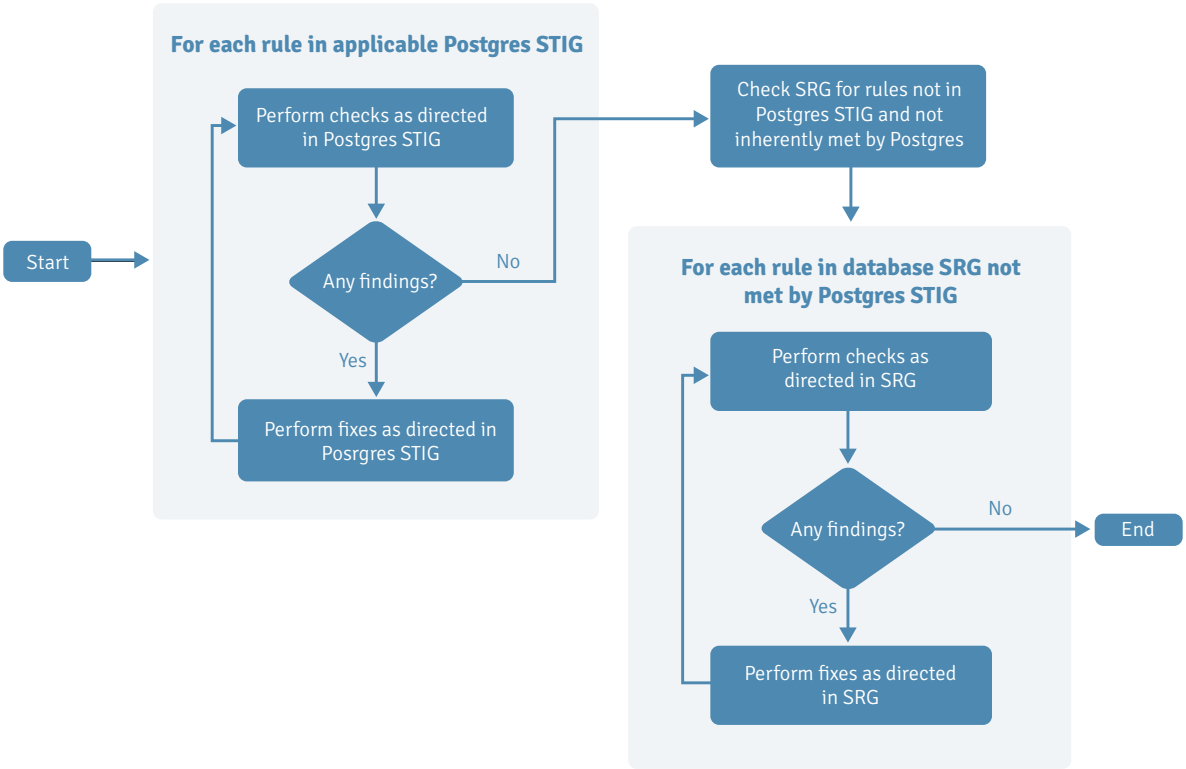


Figure 3 - General Process for Using a STIG to Secure a Postgres Database

Why There Are More Rules in the Database SRG Than in the EPAS STIGs

When developing the STIG with DISA, a vendor can identify SRG rules that are inherently met by the product; in other words, those that do not require an additional configuration. The remaining rules are identified as either capable of being met by configuring the product and related components or are not able to be met by the product. DISA only includes the rules that are configurable in a product STIG.

In the course of developing the EPAS STIGs, EDB identified that 17 of the rules in the SRG were inherently met by EPAS on RHEL and 13 were inherently met by EPAS on Windows. In addition, to make it easier to apply the EPAS STIG on Windows, the DISA team consolidated a number of the rules into a single STIG rule since the check and fix statements were the same. All the SRG rules that were satisfied by a single STIG rule are noted in the vulnerability discussion of the STIG rule. This action resulted in 34 fewer rules in the EPAS on Windows STIG. There are also some SRG rules that were expanded into two separate STIG rules each. The EPAS on RHEL STIG has six (12 total after expansion) such rules and the EPAS on Windows STIG has five (10 total after expansion). Finally, there are a few SRG rules that were added after the STIGs were created. All this explains the reason why the total number of rows in the EPAS STIGs doesn't match the total rows in the SRG. For more details or further explanation, don't hesitate to reach out to us here at EDB.

DoD organizations that are using a STIG can contact the DISA STIG Support team (contact information provided on the DoD Cyber Exchange site) for more information about SRG rules that have been marked as inherently met and those that have been marked as applicable but not met by the product. Since some of the information may be considered For Official Use Only (FOUO), DISA only makes some of the additional information available to those individuals who can access their system with DoD Common Access Card (CAC) based credentials.

EDB Postgres Enterprise Manager – Performance diagnostics showing different types of wait states

7 Finding the EDB Postgres STIGs and Related Artifacts

The following artifacts can be downloaded from the DISA STIG document library on the DoD Cyber Exchange site:

- Database SRG (V3R1)
- EDB Postgres Advanced Server STIG for RHEL (V2R1)
- EDB Postgres Advanced Server STIG for Windows (V2R1)
- Control Correlation Identifier (CCI) Info

The following link can be used to display the list (as downloadable links) of the latest available database related STIG documents:
https://public.cyber.mil/stigs/downloads/?_dl_facet_stigs=database

The following pages on the UCF STIG viewer site contain the list of findings (i.e., vulnerabilities) documented in the following documents:

- Database Security Requirements Guide
- EDB Postgres Advanced Server STIG (for RHEL)
- EDB Postgres Advanced Server STIG (for Windows)

8 Summary

The purpose of this white paper is to shed some light on how the Postgres Secure Technical Implementation Guides (STIGs) published by the U.S. Defense Information Systems Agency (DISA) can be used to secure a Postgres deployment.

To meet this goal, we discus the value of the STIGs, what they are, how they can be downloaded and viewed, and the information contained in them. In addition, we cover some specifics about the STIGs that have been published for EDB Postgres Advanced Server (EPAS) for both RHEL and Windows based deployments and provided links for accessing them.

Don't hesitate to contact us with any questions you may have.

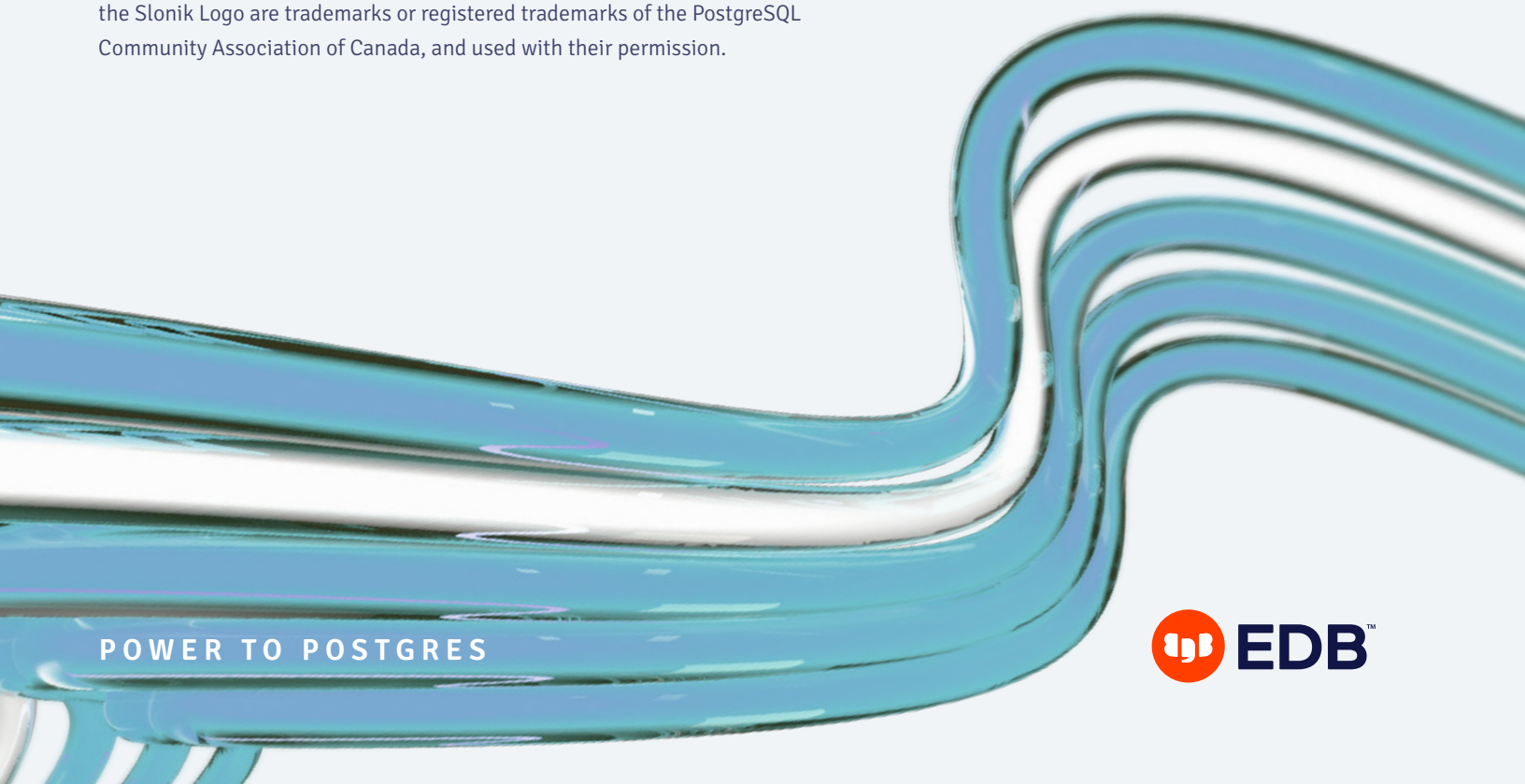
About EDB

PostgreSQL is increasingly the database of choice for organizations looking to boost innovation and accelerate business. EDB's enterprise-class software extends PostgreSQL, helping our customers get the most out of it both on premises and in the cloud. And our 24/7/365 global support, professional services, and training help our customers control risk, manage costs, and scale efficiently.

With 16 offices worldwide, EDB serves over 4,000 customers, including leading financial services, government, media and communications, and information technology organizations. To learn about PostgreSQL for people, teams, and enterprises, visit EDBpostgres.com.

How the EDB Postgres STIGs Can be Used to Secure Your PostgreSQL Database

EnterpriseDB Corporation
34 Crosby Drive
Suite 201
Bedford, MA 01730



POWER TO POSTGRES

EDB™