DbProtect AppDetective 2009.1 R2 Security Target

Version 1.0
05/16/2012

Prepared for:

Application Security, Inc.

350 Madison Avenue, 6th Floor
New York, NY 10017

Prepared By:

Science Applications International Corporation

Common Criteria Testing Laboratory

6841 Benjamin Franklin Drive
Columbia, MD 21046

LIST OF TABLES

1. Security Target Introduction

This section identifies the Security Target (ST) and Target of Evaluation (TOE) identification, ST conventions, ST conformance claims, and the ST organization. The TOE is DbProtect AppDetective 2009.1 R2 provided by Application Security, Inc. The TOE enables enterprise IT security personnel to identify and manage database vulnerabilities. With the reporting capabilities, the TOE provides a centralized console to manage security risk and extend corporate security policies at the database level.

The Security Target contains the following additional sections:

- Section 2 – Target of Evaluation (TOE) Description
  This section gives an overview of the TOE, describes the TOE in terms of its physical and logical boundaries, and states the scope of the TOE.
- Section 3 – TOE Security Environment
  This section details the expectations of the environment and the threats that are countered by the TOE and operational environment.
- Section 4 – TOE Security Objectives
  This section details the security objectives of the TOE and operational environment.
- Section 5 – IT Security Requirements
  The section presents the security functional requirements (SFR) for the TOE and details the assurance requirements.
- Section 6 – TOE Summary Specification
  The section describes the security functions represented in the TOE that satisfy the security requirements.
- Section 7 – Protection Profile Claims
  This section presents any protection profile claims.
- Section 8 – Rationale
  This section closes the ST with the justifications of the security objectives, requirements and TOE summary specifications as to their consistency, completeness, and suitability.

1.1 Security Target, TOE and CC Identification

ST Title – DbProtect AppDetective 2009.1 R2 Security Target

ST Version – Version 1.0

ST Date – 16 May 2012

TOE Identification – DbProtect AppDetective 2009.1 R2

TOE Developer – Application Security, Inc.

Evaluation Sponsor – Application Security, Inc.

CC Identification – Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 3, July 2009

1.2 Conformance Claims

This TOE is conformant to the following CC specifications:

- Common Criteria for Information Technology Security Evaluation Part 2: Security Functional Requirements, Version 3.1, Revision 3, July 2009.
  - Part 2 Extended
- Common Criteria for Information Technology Security Evaluation Part 3: Security Assurance Requirements, Version 3.1 Revision 3, July 2009.
  - Part 3 Conformant

- Assurance Level Package:
    - EAL 2 augmented with ALC_FLR.2

1.3 Conventions

The following conventions have been applied in this document:

- Security Functional Requirements – Part 2 of the CC defines the approved set of operations that may be applied to functional requirements: iteration, assignment, selection, and refinement.
    - Iteration: allows a component to be used more than once with varying operations. In the ST, iteration is indicated by a letter placed at the end of the component. For example FDP_ACC.1a and FDP_ACC.1b indicate that the ST includes two iterations of the FDP_ACC.1 requirement, a and b.
    - Assignment: allows the specification of an identified parameter. Assignments are indicated using bold and are surrounded by brackets (e.g., [**assignment**]). Note that an assignment within a selection would be identified in italics and with embedded bold brackets (e.g., [***[selected-assignment]***]).
    - Selection: allows the specification of one or more elements from a list. Selections are indicated using bold italics and are surrounded by brackets (e.g., [***selection***]).
    - Refinement: allows the addition of details. Refinements are indicated using bold, for additions, and strike-through, for deletions (e.g., "… **all** objects …" or "… ~~some~~ **big** things …").
- Other sections of the ST – Other sections of the ST use bolding to highlight text of special interest, such as captions.
- Requirements that are Part 2 extended, i.e., are not included in the CC Part 2 but are explicitly defined by the ST author, are marked with an (EXP) after the requirement name.

2. TOE Description

The Target of Evaluation (TOE) is DbProtect AppDetective 2009.1 R2. The TOE comprises a single management console component (the Console) and one or more database scanning engines (the Scan Engine). The Scan Engine component is based on the scanning engine incorporated in AppDetective Pro, which has been separately evaluated. Each of these components is an application designed to run in the context of a commercial operating system. Note that while the product comes with an Application Security ASAP Updater tool, the use of that tool is outside the scope of the evaluation. The primary reason is that the ASAP Updater is designed to keep the product up to date while the evaluation was conducted using a specific version of the product.

Note that unlike AppDetective Pro, which is designed to be directly managed (i.e., via its own application interfaces), AppDetective is designed to be managed via its associated Console component. As such, the Console component presents the administrative interfaces while each of the AppDetective Scan Engines is responsible to perform the database scanning functions.

The Scan Engine is a network-based vulnerability assessment application that reports on the security strength of database management systems (also known as database applications) within the network. The Scan Engine helps to identify vulnerable databases residing within the network by scanning for potential security vulnerabilities within those databases. The Scan Engine runs a defined set of “jobs” including Discovery jobs that discover databases on the network, Pen Test jobs that run penetration tests on the discovered databases on the network, Audit jobs that identify vulnerabilities on the discovered databases on the network, and Report jobs that provide reports on the data collected by the other three types of jobs. Administrators can then take appropriate remedial actions.

The Console is a GUI front-end that centralizes the management of multiple Scan Engines and access to the data collected by the Scan Engines. The Console enables access to set up and run AppDetective jobs. Access to AppDetective functions is role-based and can be limited to specific users based on role assignments within the Console. In addition to roles, the console also manages users and functionality in terms of groups and organizations. Groups are Windows Active Directory groups of zero to many userIDs that are assigned to organizations. Organizations are a defined set of network addresses that can be scanned by assigned userIDs and Groups. Both userIDs and Groups may be assigned to one or more Organizations. By assigning users specific roles it is possible to limit access to AppDetective functions and data. Organizations limit the scope of AppDetective Jobs to a defined set of network addresses and userIDs and Groups assigned to those Organizations can only run Jobs within the network scope assigned to that Organization.

The product includes a number of tools that are executed directly from the underlying operating system rather than from the Console including the Configuration Manager, DbProtect Migration, ASAP Updater, and Policy Editor tools. These tools are not included in the evaluated configuration (and hence are not part of the TOE). The Configuration Manager tool provides a means for modifying various configuration parameters on the Console’s host machine. The DbProtect Migration tool provides a means to migrate data from AppDetective Pro to DbProtect AppDetective. The ASAP Updater upgrades product components to the most current version. The Policy Editor provides an interface for managing policies that define the checks to be performed by Audit and Pen Test jobs. Access to these tools from the host cannot be controlled or monitored by the TOE and as such are excluded from the scope of evaluation.

2.1 TOE Overview

The TOE consists of software applications that run in the context of a commercial operating system. AppDetective discovers network accessible database applications within an organization’s infrastructure and scans them for potential vulnerabilities. AppDetective utilizes a library of known vulnerabilities and misconfiguration signatures. AppDetective includes modules that support scanning of the following database applications: Oracle, Microsoft SQL Server; IBM DB2; Sybase Adaptive Server Enterprise (ASE); MySQL; Lotus Domino; and Oracle Application Server.

In addition, AppDetective can generate fix scripts, customized based on scan results, which the administrator[1] can review and apply to address identified vulnerabilities. However, the capabilities of fix scripts have not been assessed as part of the evaluation.

AppDetective performs the following operations:

- Discovery—systematically searches the network, inventorying applications and relevant application components by vendor and release.

- Penetration Tests (Pen Tests)—applies a series of detailed security tests. AppDetective Pen Tests identify how an intruder or unauthorized user might gain access to application components. Pen Tests use various mechanisms to simulate how an intruder could exploit vulnerabilities to break into applications from the outside without possessing any authentication credentials.

- Audits— connects to the target database application and its underlying operating system to perform an assessment of its configuration, determining susceptibility to internal misuse. AppDetective Audits require a valid user account on the target application in order to verify internal configuration settings.

- Reporting—provides a reporting capability that enables the administrator to generate and view various types of report that document the results of a Pen Test or Audit, identifying potential vulnerabilities, an assessment of the risk associated with a vulnerability, and recommending actions to address a vulnerability.

Pen Tests and Audits both consist of a series of security tests or checks that are grouped together in a Policy. Each security test or check targets a specific database application type and performs actions to determine if the application is susceptible to the vulnerability tested for by the check. Pen Test and Audit checks are categorized according to the type of vulnerability for which they test.

The Pen Test categories are:

- Denial of Services—these checks examine the target application for susceptibility to specific Denial of Service attacks
- Misconfigurations—these checks examine the target application for possible misconfigurations that may leave the application susceptible to attack
- Password attacks—these checks examine the target application to determine if it is vulnerable to direct password attacks, including: accounts with blank passwords; accounts with default passwords; and susceptibility to dictionary and brute-force attacks
- Vulnerabilities—these checks determines if the application is susceptible to a specific published vulnerability for that application.

The Audit categories are:

- Access Control—these checks examine the target application for potentially inappropriate or insecure access control or privilege settings on database objects
- Application Integrity—these checks determine if specific security measures (such as enabling auditing of specific events or encrypting sensitive data) have been applied in the application
- Identification/Password Control—these checks examine the target application configuration to determine if it might be vulnerable to password attacks or problems associated with user accounts (e.g., by allowing short or poorly constructed passwords).
- OS Integrity—these checks examine aspects of the OS supporting the database application to ensure they do not expose the application to attack (e.g., permissions on database files) and that the database

[1] Note that the underlying operating system is responsible for user authentication and any user that can log in and start the TOE application is a defacto administrator.

configuration does not introduce vulnerabilities into the OS (e.g., application processes running with elevated privileges)

AppDetective includes a number of built-in Audit and Pen Test Policies that represent useful collections of checks to be performed against targeted database applications. It should be noted the evaluation has not assessed the efficacy of any specific built-in policy or its compliance with any regulatory requirements implied by the policy. Rather, the evaluation has assessed the ability of the TOE to detect particular types of vulnerability to which a targeted database may be susceptible.

The administrator can also create policies, based on the built-in policies supplied with the TOE. This involves the administrator copying an existing policy and adding or removing specific checks from the set associated with the original policy.

The Console is an SSL-enabled web-browser accessible GUI tool that provides access to functions based on user roles and configured organizations. Roles within the console include super user, admin user, basic user, and view user; different functions are available from the console based on the user role. Super and admin users can define organizations, in terms of an IPv4 address range or set of IPv4 addresses on the network, that are hierarchically organized. The super and admin users can then assign test policies and users and/or groups to applicable organizations. The test policies are sets of Pen Tests or Audit Tests grouped for convenience and associated to appropriate organizations. The users and groups are defined in the associated Active Directory, every Console user must always be assigned to at least one organization, and serve to identify which users can access each organization.

The organization hierarchy supports the definition of exclusive/isolated organizations as well as organizations that inherit IP ranges and test Policies in the “parent” Organization. Users that are assigned to an organization, either directly or via a group, can access that organization and its children organizations in accordance with their assigned role.

A Windows Administrator is assigned the Console super user role at system install and the super user can assign other Console roles to Active Directory Users, create Organizations and assign Users to Organizations. Organization assignments may also be made by Windows Groups within Active Directory, which are recognized by the Console. The Console depends upon the Operating System (OS) in the operational environment to authenticate users. The Console provides a login screen that captures the UserID and Password. It sends the UserID and Password to the OS for authentication. If the OS authenticates the user, then the Console GUI is displayed. The user is granted access to specific Console functions based on the role assigned to that UserID within the Console. The user is allowed to run vulnerability scans on specific IPv4 addresses and to access data from those scans based on the Organization to which that userID is assigned. Groups are a convenient means of assigning one or more users to the same Organization. A user may be assigned to multiple groups and may be assigned to multiple Organizations. When the user runs a vulnerability scan, that scan is run only against the IPv4 addresses in the “active” Organization for that user, i.e., the Organization displayed on the console screen at the time. The active Organization also limits the user from accessing data except that data generated by the specific IPv4 addresses assigned to that Organization. The Console allows the user to switch from one Organization to another within the same user session.

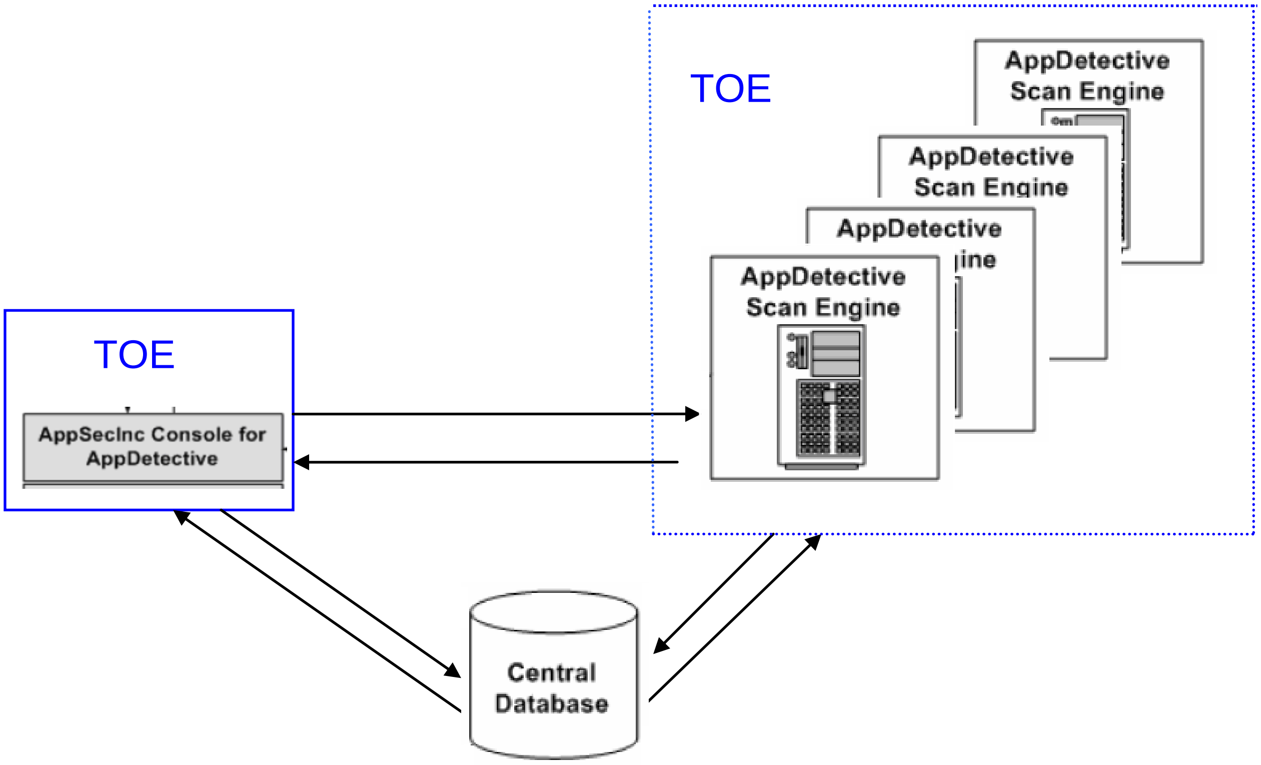
2.2 TOE Architecture

The TOE allows the authorized administrator to perform the tasks explained above (Discovery, Pen Tests, Audits) as well as examining scan results, based on function privileges.

Logically, the Scan Engine operates as a single application though it is instantiated in a series of processes utilizing inter-process communication mechanisms provided by the underlying operating system to communicate with one another. Within the host, the Scan Engine executes using the host user credentials it is configured to use.

The Console provides a means of centrally managing multiple Scan Engine instances. It implements a Graphical User Interface for the users to manage the TOE. Users can access the console remotely via a web browser. Sun Microsystems Java Runtime Environment (JRE) 1.6 is required for DbProtect Console applet to load into the web browser. Through the Console, users can access the functions and scan results of the Scan Engine instances to which they have been granted access.

The following figure depicts the association between the Console and the Scan Engine instances in their operating environment:



The TOE relies on an external Backend Database (in the operational environment) to store the scan results from AppDetective instances as well as scan policies established via the Console.

The AppDetective scan results are stored in the Backend Database, which is accessed by the Console when reporting the results. The Backend Database is in the operational Environment and is not a part of the TOE. Note that the Console and the Scan Engine depend on the underlying operating system to protect their executables and stored data images (e.g., files and registry keys), all communications between components (via an implementation of,SSL supporting HTTP and SOAP used by the TOE components), and their executing environments. The TOE also depends on the environment to ensure the Backend Database is secure.

2.2.1 Physical Boundaries

The TOE includes a DbProtect Console (including the DbProtect Console Service) and one or more instances of the AppDetective Scan Engine, which operate in the context of commercial operating systems. Each TOE component utilizes functions of their host operating system to execute, store data, and to communicate (with each other, a Backend Database to store data, and target databases) on the network as well as for security as indicated above. Note that the Backend Database could be collocated with a TOE component on a common host or could be installed on a separate host.

In addition to the components above, the product is bundled with the following components that are not included in the TOE boundary: ASAP Updater; Configuration Manager, DbProtect Migration, and Policy Editor tools. Furthermore the product is bundled some third party products that will be installed to support the TOE if not already present (see below).

The Console implements its own interface accessible to authorized users via web browsers.

Each TOE application is designed to operate in the context of the following operating systems: Microsoft Windows Server 2003 and 2008 Enterprise Edition, Microsoft Windows Server 2003 and 2008 Enterprise x64 each with the latest patches.

The TOE can be configured to use Microsoft SQL Server 2000 SP4, Microsoft SQL Server 2005, Microsoft SQL Server 2008 to store and retrieve scan results. Microsoft Access and Microsoft SQL Express are not supported in the evaluated configuration.

Additionally, the TOE requires the following components in the operational environment (and will install them if not already present upon installation of the TOE):

- Microsoft XML Core Services 4.0 SP2,
- Microsoft .NET Framework 2.0 SP1,
- Microsoft Visual Studio 2005 C++ Redistributable, and
- TomCat Engine 5.5.20.

The TOE includes Crystal Reports 9.2.0 to support its report generation and viewing function and, WinPcap Pro 4.0.2.1123 to support the Discovery operation. On the other hand, the TOE depends on WodSSH (2.4.1) in the operational environment to support the TOE’s ability to connect to target Linux/UNIX operating systems for the purpose of audits.

The TOE supports its Discovery, Pen Test and Audit operations on the following database applications:

- Oracle 11g, Oracle 10g , Oracle9i, and Oracle8i (note that while Oracle 8 and Oracle 7 are also supported, the Audit function does not work for those versions);
- Oracle Application Server 9i and 9i Release 2;
- Microsoft SQL Server Versions 6.x, 7.0, 2000, 2005, 2005 Express Edition, and 2008 and MSDE versions 1.0 and 2000 SP4;
- Lotus Domino v4.5 through 7.0;
- Sybase ASE 11.0, 11.5, 11.9.2, 12.0, 12.5, 15;
- IBM DB2 Versions 6.1, 7.1, 8.1 and 8.2;
- IBM DB2 zSeries Versions 7 and 8;
- MySQL 3.20, 3.21, 3.22, 3.23, 4.0, 4.1, and 5.0.

However, in order to perform audits on some database applications, the administrator needs to ensure the following components are installed and accessible in the operational environment:

- IBM DB2 Server audits require the IBM DB2 runtime client;
- IBM DB2 for Mainframe audits require IBM DB2 Connect;
- Lotus Domino audits require the Lotus Notes client driver; and,
- Sybase ASE audits require the Sybase ASE ODBC driver.

2.2.2 Logical Boundaries

This section identifies: the security functions provided by DbProtect AppDetective; functions provided by the operational environment in which DbProtect AppDetective operates; and DbProtect AppDetective functions not covered by the evaluation.

2.2.2.1 Evaluated Security Functions

This section addresses the following security functions provided by the TOE:

- Database Discovery and Scanning
- Security audit

- Identification and authentication
- Security management

2.2.2.1.1 Database Discovery and Scanning

The TOE is a network-based, vulnerability assessment scanner, which discovers database applications within the network infrastructure and assesses their security strength. Without requiring any agents on the target systems, the TOE can perform audits and simulate attacks against discovered and targeted applications to uncover security vulnerabilities and misconfigurations. The TOE performs the following operations on database applications:

- Discovery—systematically searches the network, inventorying applications and relevant application components by vendor and release
- Pen Test—through a series of detailed security tests and Pen Tests, the TOE identifies how an intruder or unauthorized user might gain access to application components. Pen Tests use a black-box approach to simulate how an intruder would exploit vulnerabilities to break into a database application from the outside (this is termed “outside-in” in the product documentation).
- Audit—in contrast, an Audit makes use of privileged accounts on the target database application and its underlying operating system host to determine susceptibility to internal misuse (this is termed “inside-out” in the product documentation). Audits require a valid user account in order to verify internal configuration settings.
- Report—the TOE can generate reports in various formats (HTML, XML, ASCII text) that identify specific potential vulnerabilities, provide an assessment of the risk associated with a vulnerability, and recommend actions to address a vulnerability.

2.2.2.1.2 Security audit

The TOE has the ability to generate audit records for the TOE security-relevant events. Section 6.1.2 specifies the TOE security events. The TOE records within each audit record at least the following information: Date, Time, Event Type, Success or Failure, User ID of the user. The TOE relies on the operational environment to protect and store the audit records, provide the ability to review the audit records, and to provide a reliable timestamp.

2.2.2.1.3 Identification and authentication

The Console maintains four different roles (super user, admin user, basic user and view user). The Console requires each user to provide a username and password before he/she can access any other Console security functions. The Console will pass the provided username and password to the underlying operating system and will deny the user session if the operating system does not indicate successful authentication of the username.

Note that the TOE does not authenticate users, but rather relies on its host operating system to protect it from inappropriate user access. The TOE depends upon Windows Active Directory for user information, including UserID and Windows Group. The TOE maintains UserID, Role, and Organization assignments for each user.

2.2.2.1.4 Security management

The Console provides security management functions that are accessible via an SSL-enabled web browser. Each identified user is required to be authenticated using services of the operational environment (i.e., host operating system).

The Console implements role-based access control features. As such, the Console restricts users access the management functions. There are four Console user types: Super User, admin user, basic user, and view user. The Console partitions access in two ways, by organization and by role. Organizations are created by super users. The admin user defines the users, policy access rights, and data partitioning within each organization. The main principle is to restrict data stored, collected, and associated with an organization to users in that organization. This applies to all jobs, reports, and discovery results. Organizations are hierarchically organized, which affects the visibility of one organization to another. The admin users can create new organizations or delete organizations within the organization that they have access. However, they are restricted from creating or deleting organizations above the organization to which they have access.

2.2.2.2 Functions Provided by the TOE Operational Environment

The TOE relies on the operational environment in which it operates for the following security and other functionality:

- Protect the TOE's stored executable image and its execution environment;
- Protect TOE stored data, including audit records and scan results;
- Provide a means to audit attempts to access the TOE stored executable image and stored data from the operational environment (i.e., not through the TOE's own interfaces);
- Provide a reliable time stamp for use in audit records and scan results;
- Authenticate authorized administrators and restrict the ability to manage and operate the TOE to authorized administrative users;
- Provide a means for administrators to review the audit records in the audit trail; and
- Provide encryption services used to encrypt database credentials and also to encrypt communication channels between the TOE components and also between the TOE Console and web browsers used to access it.

Additionally, the TOE relies on its host to facilitate communication with target database applications and operating system products for the purposes of scanning and auditing.

- For database applications, the TOE uses the ODBC, Oracle Instant client, DB2 client, Lotus Notes Domino C++, or TCP/IP socket APIs.
- For Windows operating systems, the TOE uses the remote registry APIs, SMB file share APIs, and Windows Management Instrumentation (WMI) APIs.
- For Linux/UNIX operating systems, the TOE uses telnet and SSH (via a third party WodSSH component).

2.2.2.3 Functions not Addressed by the Evaluation

The TOE can generate fix scripts that the administrator can apply to correct problems identified by Pen Tests or Audits. The evaluation has not covered the efficacy of these fix scripts in actually correcting detected problems.

The product provides a tool, ASAP Updater, which can be used to update the TOE and its knowledge base of application problems. However, the developer's deployment methodology is to make only complete releases of the TOE software available to customers. Use of ASAP Updater would take the TOE out of its evaluated configuration, and so it is excluded from the evaluation.

Similarly, the product includes Configuration Manager and DbProtect Migration tools. The Configuration Manager tool provides a means for modifying various configuration parameters on the Console's host machine. The DbProtect Migration tool provides a means to migrate data from AppDetective Pro to DbProtect AppDetective. These tools are not necessary for the normal use of the TOE and have been excluded from the evaluated configuration as a result.

The product also provides a Policy Editor that offers an interface for managing policies that define the checks to be performed by Audit and Pen Test jobs. Access to the Policy Editor cannot be controlled or monitored by the TOE and as such its use has not been included in the scope of evaluation, but should have no affect on the TOE itself. Rather the evaluation addresses the use of primitives found in policies however they may have been created (built-in, user created, or otherwise obtained).

The TOE provides the capability for users to create their own tests and checks for Pen Tests and Audits. However, the evaluation is unable to make any comment on the efficacy of those tests and checks not provided as part of the TOE.

Additionally, the following capabilities have been excluded from the scope of analysis during the evaluation: ability to log to a Check point event logging server; SCAP support; use of NMAP files in performing Discoveries; and CVE compatibility.

Finally, the DbProtect console component provides the ability to manage both the AppDetective product and the sibling AppRadar product, which was separately evaluated. While this TOE doesn't address the integration of both products, the use of the AppRadar product with the DbProtect console in addition to the AppDetective product does not invalidate the evaluated configuration of the TOE.

2.3 TOE Documentation

Application Security, Inc. offers a series of documents that describe the installation of the Console and AppDetective as well as guidance for subsequent use and administration of the applicable security features as follows:

- DbProtect AppDetective 2009.1 R2 Evaluated Configuration, Version 0.9, April 6, 2012
- DbProtect 2009.1R2 Administrators' Guide, April 17, 2009 (updated April 6, 2012)
- DbProtect 2009.1R2 Installation Guide, April 7, 2009
- DbProtect 2009.1R2 Users' Guide, April 21, 2009 (updated April 6, 2012)

3. Security Environment

The TOE security environment describes the security aspects of the intended environment in which the TOE is to be used and the manner in which it is expected to be employed. The statement of the TOE security environment defines the following:

- Organizational Policies that the TOE and the environment of the TOE fulfill
- Threats that the TOE and the environment of the TOE counters
- Assumptions made about the operational environment and the intended method of use for the TOE

Furthermore, the TOE is intended to be used in environments where the relative assurance that its security functions are enforced is commensurate with EAL2 as defined in the CC.

3.1 Organizational Policies

P.ACCACT Users of the TOE shall be accountable for their actions.

P.DETECT Configuration and vulnerability information that might be indicative of the potential for a future intrusion of a targeted IT System (database) must be collected.

P.MANAGE The TOE shall only be managed by authorized users.

3.2 Threats

T.COMDIS An unauthorized user may attempt to disclose the data collected by the TOE by bypassing a TOE security mechanism.

T.COMINT An unauthorized user may attempt to compromise the integrity of the data collected by the TOE by bypassing a security mechanism.

T.FACCNT Unauthorized attempts to access TOE data or security functions may go undetected.

T.IMPCON An unauthorized user may inappropriately change the configuration of the TOE causing potential vulnerability to go undetected.

T.LOSSOF An unauthorized user may attempt to remove or destroy data collected by the TOE.

T.NOHALT An unauthorized user may attempt to compromise the continuity of the collection functionality by halting execution of the TOE.

T.PRIVIL A user may gain access unauthorized to TOE security functions and data.

T.SCNCFG Improper security configuration settings may exist in the IT system the TOE monitors.

T.SCNVUL Vulnerabilities may exist in the targeted IT System the TOE monitors.

3.3 Assumptions

A.ACCESS The TOE has access to all the IT System data it needs to perform its functions.

A.DYNMIC The TOE will be managed in a manner that allows it to appropriately address changes in the IT System the TOE monitors.

A.LOCATE The processing resources of the TOE will be located within controlled access facilities, which will prevent unauthorized physical and logical access.

A.MANAGE There will be one or more competent individuals assigned to manage the TOE and the security of the information it contains.

A.NOEVIL The authorized administrators are not careless, willfully negligent, or hostile, and will follow and abide by the instructions provided by the TOE documentation.

4. Security Objectives

This section defines the security objectives for the TOE and its supporting operational environment. The security objectives are intended to counter identified threats, comply with defined organizational security policies, and address applicable assumptions.

4.1 Security Objectives for the TOE

O.ACCESS The TOE must allow authorized users to access only appropriate TOE functions and data.

O.AUDITS The TOE must be able to record audit records for data accesses and use of the primary TOE functions.

O.EADMIN The TOE must include a set of functions that allow effective management of its functions and data.

O.IDACTS The TOE must collect and store configuration and vulnerability information that might be indicative of the potential for a future intrusion of an IT System.

O.IDENT The TOE must be able to identify users prior to allowing access to the TOE.

O.PROTCT The TOE must protect itself from unauthorized modifications and access to its functions and data via its own interfaces.

4.2 Security Objectives for the TOE Operational Environment

The following security objectives for the operational environment of the TOE must be satisfied in order for the TOE to fulfill its own security objectives.

OE.AUDIT The operational environment of the TOE can audit attempts to access the TOE's stored executable image and stored data.

OE.CREDEN Those responsible for the TOE must ensure that all access credentials are protected by the users in a manner which is consistent with IT security.

OE.IDAUTH The operational environment of the TOE authenticates users prior to allowing access to TOE via its own interfaces.

OE.INSTAL Those responsible for the TOE must ensure that the TOE is delivered, installed, managed, and operated in a manner which is consistent with the TOE guidance.

OE.INTROP The TOE is interoperable with the IT System it monitors and scans.

OE.PERSON Personnel working as authorized administrators shall be carefully selected and trained for proper operation of the TOE.

OE.PHYCAL Those responsible for the TOE must ensure that those parts of the TOE critical to security policy are protected from any physical attack.

OE.PROTECT The operational environment of the TOE must protect the TOE from logical attacks including unauthorized modifications and access to stored data or TOE executables.

OE.REVIEW The operational environment of the TOE provides the capability to review the audit records.

OE.TIME The operational environment will provide reliable timestamps to the TOE.

OE.TRANSMIT The operational environment must protect the data transmitted between the TOE components and the operational environment components.

5. IT Security Requirements

The security requirements for the TOE have been drawn from Parts 2 and 3 of the Common Criteria, with the exception of some extended security functional requirements crafted to better represent the vulnerability scanning functions of the TOE. The security functional requirements have been selected to correspond to the actual security functions implemented by the TOE while the assurance requirements have been selected to offer a low to moderate level of assurance that those security functions are properly realized.

5.1 Extended Security Functional Requirements

A class of ADP (named after the AppDetective Product) security functional requirements was created to address specifically the data collected and analyzed by the TOE. The audit class of FAU was initially used as a model for creating these requirements. The purpose of this class of requirements is to address the unique nature of the DB Scan data and provide for requirements about collecting, reviewing and managing the data. These requirements embody all the necessary security functions of collection, storage, and review as well as protection of security credentials entrusted to the TOE. The TOE requires the time stamp (FPT_STM.1) to record its finding. Also, ADP_RDR_EX.1 is dependant upon ADP_SCN_EX.1 since data can be reviewed only after it is collected. ADP_PRT_EX.1 has been created explicitly to ensure that the security credentials used to access target IT systems must be protected; otherwise the targets might be more vulnerable due to the TOE. It has been added to the ADP class since it is directly related to the main scanning functions of the TOE. Each is defined as follows:

5.1.1 DB Scan Data Review (ADP_RDR)

Family Behavior

The family of SFRs is intended to address functions related to creating and reviewing scan reports created by the TOE.

Management: ADP_RDR_EX.1

The following actions could be considered for the management functions in FMT:

a) Management of the reporting function.

Audit: ADP_RDR_EX.1

The following actions should be auditable if ADP_RDR_EX.1 is included in the PP/ST:

- Minimal: Review of scan data.

5.1.1.1 DB Scan Data Review (EXP) (ADP_RDR_EX.1)

Hierarchical to: No other components

Dependencies: ADP_SCN_EX.1

ADP_RDR_EX.1.1 The TSF shall provide the authorized user with the capability to create and review DB scan reports based on DB scan data analytical results produced by the TSF.

ADP_RDR_EX.1.2 The TSF shall provide the DB scan reports in a manner suitable for the user to interpret the information.

5.1.2 DB Scan Data Collection and Analysis (ADP_SCN)

Family Behavior

The family of SFRs is intended to address functions related to scanning or collection and analysis target systems in the operational environment.

Management: ADP_SCN_EX.1

The following actions could be considered for the management functions in FMT:

a) Management of the scanning or collection function.

Audit: ADP_SCN_EX.1

The following actions should be auditable if ADP_SCN_EX.1 is included in the PP/ST:

- Minimal: Initiation of scans.

5.1.2.1 DB Scan Data Collection and Analysis (ADP_SCN_EX.1)

Hierarchical to: No other components

Dependencies: FPT_STM.1

ADP_SCN_EX.1.1 The TSF shall be able to perform identification of targeted IT system resources.

ADP_SCN_EX.1.2 The TSF shall be able to perform signature analysis[2] of identified targeted IT system resources.

ADP_SCN_EX.1.3 The TSF shall be able to collect the following information from identified targeted IT System resource(s): access control configuration, authentication configuration, accountability policy configuration, detected known vulnerabilities.

5.1.3 DB Scan Credential Protection (ADP_PRT)

Family Behavior

The family of SFRs is intended to address functions related to storing and protecting credentials used by the TOE to access scan targets in the operational environment.

Management: ADP_PRT_EX.1

The following actions could be considered for the management functions in FMT:

a) Management of the security credentials.

Audit: ADP_PRT_EX.1

There are no auditable events foreseen.

5.1.3.1 DB Scan Credential Protection (EXP) (ADP_PRT_EX.1)

Hierarchical to: No other components

Dependencies: None

ADP_PRT_EX.1.1 The TSF shall ensure that configured IT System security credentials are stored in a secure manner.

5.2 TOE Security Functional Requirements

The following table describes the SFRs satisfied by the TOE. Extended requirements are marked with (EXP) to indicate that they are not drawn from the Common Criteria.

Requirement Class	Requirement Component
ADP: DB Scan (EXP)	ADP_RDR_EX.1: DB Scan Data Review (EXP)
	ADP_SCN_EX.1: DB Scan Data Collection and Analysis (EXP)
	ADP_PRT_EX.1: DB Scan Credential Protection (EXP)
FAU: Security audit	FAU_GEN.1: Audit data generation

[2] In this context 'signature analysis' is meant to indicate that the TOE is configured with specific data or parameters used to match some analytical target (e.g., a resource such as a file).

FIA: Identification and authentication	FIA_ATD.1: User attribute definition
	FIA_UID.1: Timing of identification
FMT: Security management	FMT_MOF.1: Management of security functions behaviour
	FMT_SMF.1: Specification of Management Functions
	FMT_SMR.1: Security roles

Table 1 TOE Security Functional Components

5.2.1 DB Scan (EXP) (ADP)

5.2.1.1 DB Scan Data Review (EXP) (ADP_RDR_EX.1)

ADP_RDR_EX.1.1 The TSF shall provide the authorized user with the capability to create and review DB scan reports based on DB scan data analytical results produced by the TSF.

ADP_RDR_EX.1.2 The TSF shall provide the DB scan reports in a manner suitable for the user to interpret the information.

5.2.1.2 DB Scan Data Collection and Analysis (ADP_SCN_EX.1)

ADP_SCN_EX.1.1 The TSF shall be able to perform identification of targeted IT system resources.

ADP_SCN_EX.1.2 The TSF shall be able to perform signature analysis of identified targeted IT system resources.

ADP_SCN_EX.1.3 The TSF shall be able to collect the following information from identified targeted IT System resource(s): access control configuration, authentication configuration, accountability policy configuration, detected known vulnerabilities.

5.2.1.3 DB Scan Credential Protection (EXP) (ADP_PRT_EX.1)

ADP_PRT_EX.1.1 The TSF shall ensure that configured IT System security credentials are stored in a secure manner.

5.2.2 Security audit (FAU)

5.2.2.1 Audit data generation (FAU_GEN.1)

FAU_GEN.1.1 The TSF shall be able to generate an audit record of the following auditable events:
a) Start-up and shutdown of the audit functions;
b) All auditable events for the [***not specified***] level of audit; and
c) [
initiation of Discovery, Pen Test, or Audit scans,
scheduling Discovery, Pen test, or Audit scans,
creation and review of DB Scan reports based on DB scan data].

FAU_GEN.1.2 The TSF shall record within each audit record at least the following information:
a) Date and time of the event, type of event, subject identity (if applicable), and the outcome (success or failure) of the event; and
b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [**no additional information**].

5.2.3 Identification and authentication (FIA)

5.2.3.1 User attribute definition (FIA_ATD.1)

FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual users: [**user ID, role, organization**].

5.2.3.2 Timing of identification (FIA_UID.1)

FIA_UID.1.1 The TSF shall allow [**no functions**] on behalf of the user to be performed before the user is identified.

FIA_UID.1.2 The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

5.2.4 Security management (FMT)

5.2.4.1 Management of security functions behaviour (FMT_MOF.1)

FMT_MOF.1.1 The TSF shall restrict the ability to [***modify the behaviour of***] the functions [**defined in the table below**] to [**the user roles defined in the table below**].

Functions	Super user	Admin user	Basic user	View user
Register an AppDetective Scan Engine	access	no access	no access	no access
Unregister an AppDetective Scan Engine	access	no access	no access	no access
Edit an AppDetective Scan Engine	access	no access	no access	no access
Export credentials to a file via a Job template or Credential Profile	access	no access	no access	no access
Import credentials from file via a Job template or Credential Profile	access	access	access	no access
Create or modify a Credential Profile	access	access	access	no access
Select a different effective Organization	access	access	no access	no access
Add, edit and remove an Organization	access	access	no access	no access
Add, edit and remove a User or Group	access	access	no access	no access
Create, edit, delete, schedule, manage, or cancel a Job	access	access	access	no access
Add, edit, or remove an application to the Discovery	access	access	access	no access
Delete a Report	access	access	access	no access
Create, edit, or schedule a Report	access	access	access	access
View Reports, Job Results, and Job History	access	access	access	Access
Add an application to the Dashboard	access	access	access	No access
Edit or remove an application from the Dashboard	access	access	no access	no access

5.2.4.2 Specification of Management Functions (FMT_SMF.1)

FMT_SMF.1.1 The TSF shall be capable of performing the following security management functions: [
initiation of Discovery, Pen Test, or Audit scan jobs,
import, export, creation, and modification of Credential Profiles
scheduling Discovery, Pen test, or Audit scan jobs,
creation, removal, or modification of scan jobs,
creation and review of DB Scan reports and results based on DB scan data,
Add an application to the Dashboard
Edit or remove an application from the Dashboard
addition, removal, or modification of scan engines,
creation, removal, or modification of organizations, and
creation, removal, or modification (including organization and role assignments) of users].

5.2.4.3 Security roles (FMT_SMR.1)

FMT_SMR.1.1 The TSF shall maintain the roles [**super user, admin user, basic user, view user**].
FMT_SMR.1.2 The TSF shall be able to associate users with roles.

5.3 TOE Security Assurance Requirements

The security assurance requirements for the TOE are the EAL 2 augmented with ALC_FLR.2 components as specified in Part 3 of the Common Criteria.

Requirement Class	Requirement Component
ADV: Development	ADV_ARC.1: Security architecture description
	ADV_FSP.2: Security-enforcing functional specification
	ADV_TDS.1: Basic design
AGD: Guidance documents	AGD_OPE.1: Operational user guidance
	AGD_PRE.1: Preparative procedures
ALC: Life-cycle support	ALC_CMC.2: Use of a CM system
	ALC_CMS.2: Parts of the TOE CM coverage
	ALC_DEL.1: Delivery procedures
	ALC_FLR.2: Flaw reporting procedures
ATE: Tests	ATE_COV.1: Evidence of coverage
	ATE_FUN.1: Functional testing
	ATE_IND.2: Independent testing - sample
AVA: Vulnerability assessment	AVA_VAN.2: Vulnerability analysis

Table 2 EAL 2 augmented with ALC_FLR.2 Assurance Components

5.3.1 Development (ADV)

5.3.1.1 Security architecture description (ADV_ARC.1)

ADV_ARC.1.1d
The developer shall design and implement the TOE so that the security features of the TSF cannot be bypassed.

ADV_ARC.1.2d
The developer shall design and implement the TSF so that it is able to protect itself from tampering by untrusted active entities.

ADV_ARC.1.3d
The developer shall provide a security architecture description of the TSF.

ADV_ARC.1.1c

The security architecture description shall be at a level of detail commensurate with the description of the SFR-enforcing abstractions described in the TOE design document.

ADV_ARC.1.2c

The security architecture description shall describe the security domains maintained by the TSF consistently with the SFRs.

ADV_ARC.1.3c

The security architecture description shall describe how the TSF initialisation process is secure.

ADV_ARC.1.4c

The security architecture description shall demonstrate that the TSF protects itself from tampering.

ADV_ARC.1.5c

The security architecture description shall demonstrate that the TSF prevents bypass of the SFR-enforcing functionality.

ADV_ARC.1.1e

The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.1.2 Security-enforcing functional specification (ADV_FSP.2)

ADV_FSP.2.1d

The developer shall provide a functional specification.

ADV_FSP.2.2d

The developer shall provide a tracing from the functional specification to the SFRs.

ADV_FSP.2.1c

The functional specification shall completely represent the TSF.

ADV_FSP.2.2c

The functional specification shall describe the purpose and method of use for all TSFI.

ADV_FSP.2.3c

The functional specification shall identify and describe all parameters associated with each TSFI.

ADV_FSP.2.4c

For each SFR-enforcing TSFI, the functional specification shall describe the SFR-enforcing actions associated with the TSFI.

ADV_FSP.2.5c

For each SFR-enforcing TSFI, the functional specification shall describe direct error messages resulting from processing associated with the SFR-enforcing actions.

ADV_FSP.2.6c

The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.

ADV_FSP.2.1e

The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_FSP.2.2e

The evaluator shall determine that the functional specification is an accurate and complete instantiation of the SFRs.

5.3.1.3 Basic design (ADV_TDS.1)

ADV_TDS.1.1d

The developer shall provide the design of the TOE.

ADV_TDS.1.2d

The developer shall provide a mapping from the TSFI of the functional specification to the lowest level of decomposition available in the TOE design.

ADV_TDS.1.1c

The design shall describe the structure of the TOE in terms of subsystems.

ADV_TDS.1.2c

The design shall identify all subsystems of the TSF.

ADV_TDS.1.3c

The design shall describe the behaviour of each SFR-supporting or SFR-non-interfering TSF subsystem in sufficient detail to determine that it is not SFR-enforcing.

ADV_TDS.1.4c

The design shall summarise the SFR-enforcing behaviour of the SFR-enforcing subsystems.

ADV_TDS.1.5c

The design shall provide a description of the interactions among SFR-enforcing subsystems of the TSF, and between the SFR-enforcing subsystems of the TSF and other subsystems of the TSF.

ADV_TDS.1.6c

The mapping shall demonstrate that all TSFIs trace to the behaviour described in the TOE design that they invoke.

ADV_TDS.1.1e

The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_TDS.1.2e

The evaluator shall determine that the design is an accurate and complete instantiation of all security functional requirements.

5.3.2 Guidance documents (AGD)

5.3.2.1 Operational user guidance (AGD_OPE.1)

AGD_OPE.1.1d

The developer shall provide operational user guidance.

AGD_OPE.1.1c

The operational user guidance shall describe, for each user role, the user-accessible functions and privileges that should be controlled in a secure processing environment, including appropriate warnings.

AGD_OPE.1.2c

The operational user guidance shall describe, for each user role, how to use the available interfaces provided by the TOE in a secure manner.

AGD_OPE.1.3c

The operational user guidance shall describe, for each user role, the available functions and interfaces, in particular all security parameters under the control of the user, indicating secure values as appropriate.

AGD_OPE.1.4c

The operational user guidance shall, for each user role, clearly present each type of security-relevant event relative to the user-accessible functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_OPE.1.5c

The operational user guidance shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

AGD_OPE.1.6c

The operational user guidance shall, for each user role, describe the security measures to be followed in order to fulfil the security objectives for the operational environment as described in the ST.

AGD_OPE.1.7c

The operational user guidance shall be clear and reasonable.

AGD_OPE.1.1e

The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.2.2 Preparative procedures (AGD_PRE.1)

AGD_PRE.1.1d

The developer shall provide the TOE including its preparative procedures.

AGD_PRE.1.1c

The preparative procedures shall describe all the steps necessary for secure acceptance of the delivered TOE in accordance with the developer's delivery procedures.

AGD_PRE.1.2c

The preparative procedures shall describe all the steps necessary for secure installation of the TOE and for the secure preparation of the operational environment in accordance with the security objectives for the operational environment as described in the ST.

AGD_PRE.1.1e

The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AGD_PRE.1.2e

The evaluator shall apply the preparative procedures to confirm that the TOE can be prepared securely for operation.

5.3.3 Life-cycle support (ALC)

5.3.3.1 Use of a CM system (ALC_CMC.2)

ALC_CMC.2.1d

The developer shall provide the TOE and a reference for the TOE.

ALC_CMC.2.2d

The developer shall provide the CM documentation.

ALC_CMC.2.3d

The developer shall use a CM system.

ALC_CMC.2.1c

The TOE shall be labelled with its unique reference.

ALC_CMC.2.2c

The CM documentation shall describe the method used to uniquely identify the configuration items.

ALC_CMC.2.3c

The CM system shall uniquely identify all configuration items.

ALC_CMC.2.1e

The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.3.2 Parts of the TOE CM coverage (ALC_CMS.2)

ALC_CMS.2.1d

The developer shall provide a configuration list for the TOE.

ALC_CMS.2.1c

The configuration list shall include the following: the TOE itself; the evaluation evidence required by the SARs; and the parts that comprise the TOE.

ALC_CMS.2.2c

The configuration list shall uniquely identify the configuration items.

ALC_CMS.2.3c

For each TSF relevant configuration item, the configuration list shall indicate the developer of the item.

ALC_CMS.2.1e

The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.3.3 Delivery procedures (ALC_DEL.1)

ALC_DEL.1.1d

The developer shall document and provide procedures for delivery of the TOE or parts of it to the consumer.

ALC_DEL.1.2d

The developer shall use the delivery procedures.

ALC_DEL.1.1c

The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to the consumer.

ALC_DEL.1.1e

The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.3.4 Flaw reporting procedures (ALC_FLR.2)

ALC_FLR.2.1d

The developer shall document and provide flaw remediation procedures addressed to TOE developers.

ALC_FLR.2.2d

The developer shall establish a procedure for accepting and acting upon all reports of security flaws and requests for corrections to those flaws.

ALC_FLR.2.3d

The developer shall provide flaw remediation guidance addressed to TOE users.

ALC_FLR.2.1c

The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the TOE.

ALC_FLR.2.2c

The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.

ALC_FLR.2.3c

The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.

ALC_FLR.2.4c

The flaw remediation procedures documentation shall describe the methods used to provide flaw information, corrections and guidance on corrective actions to TOE users.

ALC_FLR.2.5c

The flaw remediation procedures shall describe a means by which the developer receives from TOE users reports and enquiries of suspected security flaws in the TOE.

ALC_FLR.2.6c

The procedures for processing reported security flaws shall ensure that any reported flaws are remediated and the remediation procedures issued to TOE users.

ALC_FLR.2.7c

The procedures for processing reported security flaws shall provide safeguards that any corrections to these security flaws do not introduce any new flaws.

ALC_FLR.2.8c

The flaw remediation guidance shall describe a means by which TOE users report to the developer any suspected security flaws in the TOE.

ALC_FLR.2.1e

The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.4 Tests (ATE)

5.3.4.1 Evidence of coverage (ATE_COV.1)

ATE_COV.1.1d

The developer shall provide evidence of the test coverage.

ATE_COV.1.1c

The evidence of the test coverage shall show the correspondence between the tests in the test documentation and the TSFIs in the functional specification.

ATE_COV.1.1e

The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.4.2 Functional testing (ATE_FUN.1)

ATE_FUN.1.1d

The developer shall test the TSF and document the results.

ATE_FUN.1.2d

The developer shall provide test documentation.

ATE_FUN.1.1c

The test documentation shall consist of test plans, expected test results and actual test results.

ATE_FUN.1.2c

The test plans shall identify the tests to be performed and describe the scenarios for performing each test. These scenarios shall include any ordering dependencies on the results of other tests.

ATE_FUN.1.3c

The expected test results shall show the anticipated outputs from a successful execution of the tests.

ATE_FUN.1.4c

The actual test results shall be consistent with the expected test results.

ATE_FUN.1.1e

The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.4.3 Independent testing - sample (ATE_IND.2)

ATE_IND.2.1d

The developer shall provide the TOE for testing.

ATE_IND.2.1c

The TOE shall be suitable for testing.

ATE_IND.2.2c

The developer shall provide an equivalent set of resources to those that were used in the developer's functional testing of the TSF.

ATE_IND.2.1e

The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_IND.2.2e

The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.

ATE_IND.2.3e

The evaluator shall test a subset of the TSF to confirm that the TSF operates as specified.

5.3.5 Vulnerability assessment (AVA)

5.3.5.1 Vulnerability analysis (AVA_VAN.2)

AVA_VAN.2.1d

The developer shall provide the TOE for testing.

AVA_VAN.2.1c

The TOE shall be suitable for testing.

AVA_VAN.2.1e

The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_VAN.2.2e

The evaluator shall perform a search of public domain sources to identify potential vulnerabilities in the TOE.

AVA_VAN.2.3e

The evaluator shall perform an independent vulnerability analysis of the TOE using the guidance documentation, functional specification, TOE design and security architecture description to identify potential vulnerabilities in the TOE.

AVA_VAN.2.4e

The evaluator shall conduct penetration testing, based on the identified potential vulnerabilities, to determine that the TOE is resistant to attacks performed by an attacker possessing Basic attack potential.

6. TOE Summary Specification

This chapter describes the security functions:

- Database Discovery and Scanning
- Security Audit
- Identification and Authentication
- Security Management

The TOE ensures that its mechanisms cannot be bypassed within its scope of control by requiring users to be identified and also authenticated by the operational environment. Once the user is authorized as valid user listed in Windows Active Directory, the user identity is then verified by TOE with its backend database. The backend database is accessed by both the Sensor (to store data) and the Console (to query data) using Microsoft ODBC. Access to the TOE functions, via TOE Console interface, will be granted based on the user identity and role / privilege assigned. That is, upon logged on, each user to limited by the TOE to perform only those functions they are allowed based on their role and organization. Protection of the TOE via other interfaces, such as those of the host OS, are assumed to be protected as summarized elsewhere in this ST.

Similarly, the AppDetective Scan Engine component of the TOE will interact only with the Console Service. During AppDetective Scan Engine component registration, the DbProtect Console service identifies its network interface (MAC address) and provides steps to register the AppDetective Scan Engine component through its web based GUI interface. Each AppDetective Scan Engine component can be registered only by one Console. The AppDetective Scan Engine component will communicate only with the console with which it has been registered to communicate using SOAP over HTTP/SSL (HTTPS) on designated port. The HTTPS function is implemented in the operational environment (via the available web server front end and utilized by the TOE for both Console to Sensor communication as well as Console to web-browser communication. Additionally, the TOE employs WodSSH in the operational environment to establish Telnet or SSH, as supported by the target, connections with UNIX/Linux targets when performing audits.

6.1 Database Discovery and Scanning

The TOE provides a graphical user interface (GUI) – Console service - to manage the scanning of targeted IT systems (comprising the supported database applications identified in Section 2.2.1) and the review of the analytical results it produces. The Console component of the TOE provides all the necessary management functions and communicates directly with the TOE Scan Engine components using SOAP over HTTPS and indirectly with the TOE Scan Engine components (i.e., to retrieve collected data) via a shared backend database component in the operational environment.

The TOE is capable of performing various checks on targeted network accessible database applications, according to the configured job and policy. The TOE relies on the underlying operating system to provide a reliable timestamp. In addition, the TOE uses a database in the operational environment (termed the "Backend Database") to store its scan results.

The TOE is capable of creating and scheduling the following types of jobs:

- **Discovery Jobs**: The Discovery process is a port scan of devices on the network performed by the TOE to discover target database applications. The administrator may specify that Discovery be performed for an IPv4 address range or for an entire network. A Discovery locates database applications on the network, identifies the database application's IPv4 addresses and ports used to provide network services, and saves the information for subsequent use to initiate a Pen Test or Audit. The TOE incorporates WinPcap by CACE Technologies to perform the Discovery job. WinPcap is a tool for link-layer network access in Windows environments. It provides access to the network device's TCP/IP stack. Additional information on WinPcap is available at http://www.cacetech.com/products/winpcap_professional.htm.
- **Pen Tests Jobs**: A Pen Test assesses the security of a database application by running security checks against it from the network—the TOE does not need to login to the database. These checks are signature-

based where configured signatures are compared against network discernable characteristics of the target database application. The TOE can perform the following types of security checks (i.e., includes applicable signatures) as part of a Pen Test:

  - Denial of Services—these checks examine the database application for susceptibility to specific Denial of Service attacks.
  - Misconfigurations—these checks examine the database application for possible misconfigurations that may leave the database application susceptible to attack.
  - Password attacks—these checks examine the database application to determine if it is vulnerable to password attacks, including: accounts with blank passwords; accounts with default passwords still set; susceptibility to dictionary and brute-force attacks.
  - Vulnerabilities—each of these checks determines if the database is susceptible to any specific published vulnerabilities for that database application.

- **Audit Jobs**: An Audit assesses the security of a database application by connecting to the database application using an appropriately privileged account and accessing the internal configuration. The TOE can perform the following types of security checks as part of an Audit:
  - Access Control—these checks examine the database application access control configuration for potentially inappropriate or insecure access control or privilege settings on database objects.
  - Accountability—these checks examine the database application accountability configuration to determine if specific security measures, such as enabling auditing of specific events, have been applied.
  - Authentication—these checks examine the database application authentication-related configuration to determine if it is vulnerable to password attacks or problems associated with user authentication.

Data identifying the targeted database application is stored in the Backend Database as a part of the Discovery process and this data is used by the TOE to determine which policies are applicable and to run appropriate Audit and Pen Test scans. The data returned for successful scans is dependent on the scan; specific data collected is defined for each policy and database application in the user guidance documentation.

DB Scan reports are designed to communicate vulnerabilities discovered by AppDetective. The reports present the scan results in a human readable format and can include all the information from those results**.** These reports can be generated in a number of formats. AppDetective comes with a set of pre-defined scans, which are defined by Policies that are used by Pen Test or Audit Jobs.

The scan results stored by the TOE identify the date and time the job was run, the type of job (see above) that was run, and details specific to the type and results of the scan including identifying any network applications subject to scanning and any vulnerabilities identified for scanned network applications. The user guide should be consulted for more specific information about the range and presentation of information collected and made available to TOE users.

Note that while the TOE includes a number of pre-defined policies (or signatures) to identify known vulnerabilities, it provides a limited ability for the user to extend existing policies or create new policies based on their own checks. Furthermore, no assessment has been made regarding the efficacy of the pre-defined policies nor are any user-defined policies addressed within the scope of the evaluation.

As indicated above, the TOE can access some targeted IT systems using privileged accounts in order to gather specific internal configuration data. Applicable security credentials can be configured into the TOE and the TOE will store and protect those credentials using capabilities provided by its host operating system. Specifically, the TOE calls upon the Windows Data Protection API (DPAPI) to encrypt and store the credentials in its registry so that they can be recalled by the TOE when needed.

The Database Discovery and Scanning function is designed to satisfy the following security functional requirements:

- ADP_RDR_EX.1: The TOE provides the authorized users with the capability to view the DB Scan reports.
- ADP_SCN_EX.1: The TOE can identify targeted IT systems (database applications), analyze the target using signatures in order to discover potential vulnerabilities, and collect from targets configuration data pertaining to access control, accountability, and authentication as well as information related to any identified vulnerabilities.
- ADP_PRT_EX.1: The TOE uses the data protection API of its host operating system in order to securely store security credentials used to access targeted IT systems.

6.2 Security Audit

The TOE provides its own audit mechanism that can generate audit records for the use of TOE's security functions. The TOE relies on the underlying Operating System (OS) to protect and store the audit records, provide the ability to review the audit records, and to provide a reliable timestamp.

The TOE generates the following security relevant events:

- initiation of Discovery, Pen Test, or Audit scans,
- scheduling Discovery, Pen test, or Audit scans,
- creation and review of DB Scan reports based on DB scan data,

Note that the audit mechanism is always enabled and there is no capability to disable the audit. The Evaluated Configuration Guide provides instructions to configure the DbProtect Console host to create an audit record when the TOE Console starts and upon shutdown effectively showing when audit starts up and stops. However, the TOE itself always generates audit events while it is running and auditing cannot be disabled obviating any need to audit start-up and shutdown of audit which is simply not applicable.

Each audit records is generated with the date and time queried from the operating system and the job type or the type of event per the list above. The outcome for each event is generally successful and implied by the event itself since the corresponding actions are made available to and can only be attempted by authorized users. In other words, the users are presented only with functions they are allowed to access and hence are prevented from attempting unauthorized functions.

It should be noted that since other events, such as user authentication and access to TOE files, occur in the host OS they can be audited by the OS and reviewed using OS audit review capabilities. Furthermore, while there are no claims of audit review, the events identified above are stored in the backend database and can be queried from that database for review. However, the TOE does include a limited ability to review job history where the audited events would be evident.

The Security audit function is designed to satisfy the following security functional requirements:

- FAU_GEN.1: The TOE generates audit events for data accesses and use of the TOE functions.

6.3 Identification and Authentication

The Console component of the TOE identifies its own users but also requires each user to be successfully authenticated before they are considered genuine and can access the functions available via the Console (i.e., TOE). The TOE maintains each user's identity, role, and organizations. Any authorized user (example: super user) must enter his/her unique user name and password when accessing the Console via an SSL-enabled Web Browser. If the user identity is known to the TOE (i.e., defined), the TOE passes this information to the underlying operating system and relies on the underlying operating system to authenticate the provided user identity. The TOE then uses that user identity to determine the applicable role and organizations. If the user identity is unknown to the TOE or the operating system fails to indicate successful authenticate of the user identity, the Console discontinues the attempt to create a user session.

The Identification and authentication function is designed to satisfy the following security functional requirements:

- FIA_ATD.1: The TOE maintains the list of security attributes belonging to individual users.
- FIA_UID.1: The TOE requires each user to be identified before accessing security functions via the Console interface.

6.4 Security Management

The Console implements management functions that are accessible via an SSL-enabled web-browser after being identified and authenticated as described in the previous section.

The Console component (i.e., Console service) of the TOE maintains the following four security management roles: super user, admin user, basic user, and view user and is able to associate defined users with one of these roles. The Console partitions access in two ways, by organizations (assigned to each user) and by role. Organizations are created by super users or admin users. The super users and admin users can define organizations that are descendents of the organization to which they belong. The main principle is that data stored, collected, and associated with an organization is only accessible by users in that (or a parent of that) organization.

Initially, when the Console is installed, there is just one organization, namely the 'root'. The user who installs the Console is automatically a super user and a member of the root organization. The super user may create new organizations and assign other users to them. The new organizations can only be created as a descendent of the root organization or its descendents. Once this organizational structure is in place, the admin user or super user can add users to these organizations. The users added to a specific organization can view only the data objects associated with their organization, with the exception of the super user and admin user. The super users and admin users are allowed to see the organizational structure and change the effective (i.e., their current) organization to another that is visible to them. This change is effective for the duration of their logon session or until changed again. Moreover, a super user or admin user is allowed to remove or add organizations under his/her organization (i.e., descendents), but he/she may not remove his/her own organization or user name. Super user, admin user and Basic user are also able to add applications to existing scan discovery. However, only Super admin user and admin user are allowed to remove application from the dashboard where discovered applications are listed. Note that a dashboard displays detailed data about application discovered on the network, recent Pen Tests and Audits performed, and any vulnerabilities detected.

When a user is created, his/her role is assigned from one of: super user, admin user, basic user, or view user. These roles affect the accessible system features for the user. As discussed above, the admin users may change their effective organization for the duration of their session. However, the super-admin has other rights, such as visibility to the Scan Engine configuration page. The basic user does not have access to view organization data, create organizations, nor add/remove users. These features are simply not present on his/her user interface. Users in one organization cannot see any data collected or saved in another organization, with the exception of the super user and admin user who first explicitly changes his effective organization.

The table in section 5.1.4.1 indicates the functions available to the identified roles. These functions are restricted by limiting the user interfaces to ‘see’ only those functions that they are allowed to access. If a user is limited based on the organization hierarchy, they cannot perceive organizations that are outside their scope of access. If a user is limited to specific functions, they will be provided options only to access the functions they are allowed to use.

The Security management function is designed to satisfy the following security functional requirements:

- FMT_MOF.1: The TOE restricts the TOE management functions to the authorized user. (please see table in section 5.2.4.1).
- FMT_SMF.1: The TOE is able to perform the following security management functions: management of the Console Access Control FSP, management of user accounts, management of AppDetective tasks and data.
- FMT_SMR.1: The TOE maintains super user, admin user, and basic user and view user roles.

7. Protection Profile Claims

There are no Protection Profile claims in this Security Target.

8. Rationale

This section provides the rationale for completeness and consistency of the Security Target. The rationale addresses the following areas:

- Security Objectives;
- Security Functional Requirements;
- Security Assurance Requirements;
- Requirement Dependencies; and
- TOE Summary Specification.

8.1 Security Objectives Rationale

This section shows that all secure usage assumptions, organizational security policies, and threats are completely covered by security objectives. In addition, each objective counters or addresses at least one assumption, organizational security policy, or threat.

8.1.1 Security Objectives Rationale for the TOE and Operational Environment

This section provides evidence demonstrating the coverage of organizational policies, threats, and usage assumptions by the security objectives.

	P.ACCACT	P.DETECT	P.MANAGE	T.COMDIS	T.COMINT	T.FACCNT	T.IMPCON	T.LOSSOF	T.NOHALT	T.PRIVIL	T.SCNCFG	T.SCNVUL	A.ACCESS	A.DYNMIC	A.LOCATE	A.MANAGE	A.NOEVIL
O.ACCESS			X	X	X		X	X	X	X	X	X					
O.AUDITS	X					X											
O.EADMIN			X														
O.IDACTS		X									X	X					
O.IDENT	X		X	X	X		X	X	X	X							
O.PROTCT			X	X	X			X		X							
OE.AUDIT						X											
OE.CREDEN			X														X
OE.IDAUTH			X	X	X		X	X	X	X							
OE.INSTAL			X				X										X
OE.INTROP													X	X			
OE.PERSON			X											X		X	
OE.PHYCAL															X		
OE.PROTECT								X	X	X					X		
OE.REVIEW	X					X											
OE.TIME	X																
OE.TRANSMIT				X	X												

Table 3 Environment to Objective Correspondence

8.1.1.1 P.ACCACT

Users of the TOE shall be accountable for their actions.

This Organizational Policy is satisfied by ensuring that:

- O.AUDITS: The O.AUDITS objective implements this policy by requiring auditing of scan data accesses and use of the primary IDS TOE functions.

- O.IDENT: The O.IDENT objective supports this policy by ensuring each user is uniquely identified.
- OE.REVIEW: The OE.REVIEW objective supports this policy by ensuring the operational environment of the TOE provides the capability to review the generated TOE audit records.
- OE.TIME: The OE.TIME objective supports this policy by ensuring a reliable time stamp is provided by the operational environment.

8.1.1.2 P.DETECT

Configuration and vulnerability information that might be indicative of the potential for a future intrusion of a targeted IT System (database) must be collected

This Organizational Policy is satisfied by ensuring that:

- O.IDACTS: The O.IDACTS objective addresses this policy by requiring collection of scanned configuration and vulnerability data.

8.1.1.3 P.MANAGE

The TOE shall only be managed by authorized users.

This Organizational Policy is satisfied by ensuring that:

- O.ACCESS: The O.ACCESS objective builds upon the O.IDENT and OE.IDAUTH objectives by only permitting authorized users to access TOE functions.
- O.EADMIN: the O.EADMIN objective ensures there is a set of functions for administrators to use.
- O.IDENT: The O.IDENT objective provides identification of users prior to any TOE function accesses.
- OE.IDAUTH: The OE.IDAUTH environment objective in conjunction with O.IDENT provides authentication of users prior to any TOE access.
- O.PROTCT: O.PROTCT objective addresses this policy by providing TOE self-protection.
- OE.CREDEN: The OE.CREDEN objective requires administrators to protect all authentication data.
- OE.INSTAL: The OE.INSTAL objective supports the O.PERSON objective by ensuring administrator follow all provided documentation and maintain the security policy.
- OE.PERSON: The OE.PERSON objective ensures competent administrators will manage the TOE.

8.1.1.4 T.COMDIS

An unauthorized user may attempt to disclose the data collected by the TOE by bypassing a security mechanism.

This Threat is satisfied by ensuring that:

- O.ACCESS: The O.ACCESS objective builds upon the O.IDAUTH objective by only permitting authorized users to access TOE data.
- O.IDENT: The O.IDENT objective provides identification of users prior to any TOE function accesses.
- OE.IDAUTH: The OE.IDAUTH environment objective in conjunction with O.IDENT provides authentication of users prior to any TOE access.
- O.PROTCT: The O.PROTCT objective addresses this threat by providing TOE self-protection.
- OE.TRANSMIT: The operational environment must protect the data transmitted between the TOE and operational environment components.

8.1.1.5 T.COMINT

An unauthorized user may attempt to compromise the integrity of the data collected by the TOE by bypassing a security mechanism.

This Threat is satisfied by ensuring that:

- O.ACCESS: The O.ACCESS objective builds upon the O.IDAUTH objective by only permitting authorized users to access TOE data.
- O.IDENT: The O.IDENT objective provides identification of users prior to any TOE function accesses.
- OE.IDAUTH: The OE.IDAUTH environment objective in conjunction with O.IDENT provides authentication of users prior to any TOE access.
- O.PROTCT: The O.PROTCT objective addresses this threat by providing TOE self-protection.
- OE.TRANSMIT: The operational environment must protect the data in transmit between the TOE and operational environment components.

8.1.1.6 T.FACCNT

Unauthorized attempts to access TOE data or security functions may go undetected.

This Threat is satisfied by ensuring that:

- O.AUDITS: The O.AUDITS objective counters this threat by requiring the TOE to audit attempts for data accesses and use of TOE functions.
- OE.AUDIT: The OE.AUDIT objective ensures the operational environment of the TOE does its part in ensuring there is accountability for accessing the TOE executable and data.
- OE.REVIEW: The OE.REVIEW objective supports this policy by ensuring the operational environment of the TOE provides the capability to review the generated TOE audit records.

8.1.1.7 T.IMPCON

An unauthorized user may inappropriately change the configuration of the TOE causing potential vulnerability to go undetected.

This Threat is satisfied by ensuring that:

- O.ACCESS: The O.ACCESS objective builds upon the O.IDAUTH objective by only permitting authorized users to access TOE functions and data.
- O.IDENT: The O.IDENT objective provides identification of users prior to any TOE function accesses.
- OE.IDAUTH: The OE.IDAUTH environment objective in conjunction with O.IDENT provides authentication of users prior to any TOE access.
- OE.INSTAL: The OE.INSTAL objective states the authorized administrators will configure the TOE properly.

8.1.1.8 T.LOSSOF

An unauthorized user may attempt to remove or destroy data collected by the TOE.

This Threat is satisfied by ensuring that:

- O.ACCESS: The O.ACCESS objective builds upon the O.IDAUTH objective by only permitting authorized users to access TOE data.
- O.IDENT: The O.IDENT objective provides identification of users prior to any TOE function accesses.
- OE.IDAUTH: The OE.IDAUTH environment objective in conjunction with O.IDENT provides authentication of users prior to any TOE access.
- O.PROTCT: The O.PROTCT objective addresses this threat by providing TOE self-protection.

8.1.1.9 T.NOHALT

An unauthorized user may attempt to compromise the continuity of the collection functionality by halting execution of the TOE.

This Threat is satisfied by ensuring that:

- O.ACCESS: The O.ACCESS objective builds upon the O.IDAUTH objective by only permitting authorized users to access.

- O.IDENT: The O.IDENT objective provides identification of users prior to any TOE function accesses.
- OE.IDAUTH: The OE.IDAUTH environment objective in conjunction with O.IDENT provides authentication of users prior to any TOE access.
- OE.PROTECT: The OE.PROTECT objective ensures that the environment provides a secure environment for the TOE.

8.1.1.10 T.PRIVIL

A user may gain unauthorized access to TOE security functions and data.

This Threat is satisfied by ensuring that:

- O.ACCESS: The O.ACCESS objective builds upon the O.IDAUTH objective by only permitting authorized users to access TOE functions.
- O.IDENT: The O.IDENT objective provides identification of users prior to any TOE function accesses.
- OE.IDAUTH: The OE.IDAUTH environment objective in conjunction with O.IDENT provides authentication of users prior to any TOE access.
- O.PROTCT: The O.PROTCT objective addresses this threat by providing TOE self-protection.
- OE.PROTECT: The OE.PROTECT objective ensures that the environment provides a secure environment for the TOE.

8.1.1.11 T.SCNCFG

Improper security configuration settings may exist in the IT system the TOE monitors.

This Threat is satisfied by ensuring that:

- O.ACCESS: The O.ACCESS objective builds on O.IDACTS by requiring the results of the scans to be accessible.
- O.IDACTS: The O.IDACTS objective counters this threat by requiring the TOE collect and store vulnerability information that might be indicative of a configuration setting change.

8.1.1.12 T.SCNVUL

Vulnerabilities may exist in the targeted IT System the TOE monitors.

This Threat is satisfied by ensuring that:

- O.ACCESS: The O.ACCESS objective builds on O.IDACTS by requiring the results of the scans to be accessible.
- O.IDACTS: The O.IDACTS objective counters this threat by requiring the TOE collect and store configuration and vulnerability information that might be indicative of a vulnerability.

8.1.1.13 A.ACCESS

The TOE has access to all the IT System data it needs to perform its functions.

This Assumption is satisfied by ensuring that:

- OE.INTROP: The OE.INTROP objective ensures the TOE has the needed access.

8.1.1.14 A.DYNMIC

The TOE will be managed in a manner that allows it to appropriately address changes in the IT System the TOE monitors.

This Assumption is satisfied by ensuring that:

- OE.INTROP: The OE.INTROP objective ensures the TOE has the proper access to the IT System.

- OE.PERSON: The OE.PERSON objective ensures that the TOE will managed appropriately.

8.1.1.15 A.LOCATE

The processing resources of the TOE will be located within controlled access facilities, which will prevent unauthorized physical and logical access.

This Assumption is satisfied by ensuring that:

- OE.PHYCAL: The OE.PHYCAL provides for the physical protection of the TOE.
- OE.PROTECT: The OE.PROTECT objective ensures that the environment provides protection from logical attacks.

8.1.1.16 A.MANAGE

There will be one or more competent individuals assigned to manage the TOE and the security of the information it contains.

This Assumption is satisfied by ensuring that:

- OE.PERSON: The OE.PERSON objective ensures all authorized administrators are qualified and trained to manage the TOE.

8.1.1.17 A.NOEVIL

The authorized administrators are not careless, willfully negligent, or hostile, and will follow and abide by the instructions provided by the TOE documentation.

This Assumption is satisfied by ensuring that:

- OE.CREDEN: The OE.CREDEN objective supports this assumption by requiring protection of all authentication data.
- OE.INSTAL: The OE.INSTAL objective ensures that the TOE is properly installed and operated.

8.2 Security Requirements Rationale

This section provides evidence supporting the internal consistency and completeness of the components (requirements) in the Security Target. Note that **Table 4** indicates the requirements that effectively satisfy the individual objectives.

8.2.1 Security Functional Requirements Rationale

All Security Functional Requirements (SFR) identified in this Security Target are fully addressed in this section and each SFR is mapped to the objective for which it is intended to satisfy.

	O.ACCESS	**O.AUDITS**	**O.EADMIN**	**O.IDACTS**	**O.IDENT**	**O.PROTCT**
ADP_RDR_EX.1	X		X			
ADP_SCN_EX.1				X		
ADP_PRT_EX.1						X
FAU_GEN.1		X				
FIA_ATD.1					X	
FIA_UID.1					X	
FMT_MOF.1	X					X
FMT_SMF.1			X			

FMT_SMR.1	X					

Table 4 Objective to Requirement Correspondence

8.2.1.1 O.ACCESS

The TOE must allow authorized users to access only appropriate TOE functions and data.

This TOE Security Objective is satisfied by ensuring that:

- ADP_RDR_EX.1: The TOE must provide the ability for authorized administrators to view the Scanner data collected from an IT System.
- FMT_MOF.1 (EXP): Only authorized user can perform the management functionality associated with their role as identified in FMT_MOF_EX.1.
- FMT_SMR.1: The TOE must be able to recognize the different administrative and user roles that exist for the TOE.

8.2.1.2 O.AUDITS

The TOE must record audit records for data accesses and use of the TOE functions.

This TOE Security Objective is satisfied by ensuring that:

- FAU_GEN.1: Security-relevant events (TOE data accesses and use of the TOE functions) must be defined and audited for the TOE.

8.2.1.3 O.EADMIN

The TOE must include a set of functions that allow effective management of its functions and data.

This TOE Security Objective is satisfied by ensuring that:

- ADP_RDR_EX.1: The TOE must provide the ability for authorized administrators to view the DB Scan data collected from an IT System.
- FMT_SMF.1: The TOE must be capable of performing the security management functions.

8.2.1.4 O.IDACTS

The TOE must collect and store configuration and vulnerability information that might be indicative of the potential for a future intrusion of an IT System.

This TOE Security Objective is satisfied by ensuring that:

- ADP_SCN_EX.1: The Scanner is required to collect and store vulnerability and configuration information of an IT System.

8.2.1.5 O.IDENT

The TOE must be able to identify users prior to allowing access to the TOE.

This TOE Security Objective is satisfied by ensuring that:

- FIA_UID.1: The TOE requires each user to identify itself before allowing any other TSF-mediated actions on behalf of that user.
- FIA_ATD.1: Security attributes of subjects use to enforce the authentication policy of the TOE must be defined.

8.2.1.6 O.PROTCT

The TOE must protect itself from unauthorized modifications and access to its functions and data via its own interfaces.

This TOE Security Objective is satisfied by ensuring that:

- ADP_PRT_EX.1: The TOE is required to protect credentials used to scan target IT Systems.
- FMT_MOF.1 : Only authorized user can perform the management functionality of the TOE identified in the table in FMT_MOF.1.
-

8.3 Security Assurance Requirements Rationale

EAL2 was selected as the assurance level because the TOE is a commercial product whose users require a low to moderate level of independently assured security. Application Security, DbProtect AppDetective is targeted for an environment with good physical access security and competent administrators. Within such environment, it is assumed that attackers will have little attack potential. As such, EAL2 is appropriate to provide the assurance necessary to counter the limited potential for attack. EAL2 is augmented with ALC_FLR.2, Flaw reporting procedures to provide instructions to users for reporting flaws, to provide internal vendor procedures for identifying, tracking and correcting product flaws.

8.4 Requirement Dependency Rationale

The following table identifies the dependencies of the requirements in this ST, including the extended requirements defined in this ST. As indicated in the table, all of the dependencies are satisfied, except those identified in **[bold-red-bracketed]** text.

ST Requirement	CC Dependencies	ST Dependencies
ADP_PRT_EX.1	none	none
ADP_RDR_EX.1	ADP_SCN_EX.1	ADP_SCN_EX.1
ADP_SCN_EX.1	FPT_STM.1	**[FPT_STM.1]**
FAU_GEN.1	FPT_STM.1	**[FPT_STM.1]**
FIA_ATD.1	none	none
FIA_UID.1	none	none
FMT_MOF.1	FMT_SMR.1 and FMT_SMF.1	FMT_SMR.1 and FMT_SMF.1
FMT_SMF.1	none	none
FMT_SMR.1	FIA_UID.1	FIA_UID.1
ADV_ARC.1	ADV_FSP.1 and ADV_TDS.1	ADV_FSP.2 and ADV_TDS.1
ADV_FSP.2	ADV_TDS.1	ADV_TDS.1
ADV_TDS.1	ADV_FSP.2	ADV_FSP.2
AGD_OPE.1	ADV_FSP.1	ADV_FSP.2
AGD_PRE.1	none	none
ALC_CMC.2	ALC_CMS.1	ALC_CMS.2
ALC_CMS.2	none	none
ALC_DEL.1	none	none
ALC_FLR.2	none	none
ATE_COV.1	ADV_FSP.2 and ATE_FUN.1	ADV_FSP.2 and ATE_FUN.1
ATE_FUN.1	ATE_COV.1	ATE_COV.1
ATE_IND.2	ADV_FSP.2 and AGD_OPE.1 and AGD_PRE.1 and ATE_COV.1 and ATE_FUN.1	ADV_FSP.2 and AGD_OPE.1 and AGD_PRE.1 and ATE_COV.1 and ATE_FUN.1
AVA_VAN.2	ADV_ARC.1 and ADV_FSP.2 and ADV_TDS.1 and AGD_OPE.1 and AGD_PRE.1	ADV_ARC.1 and ADV_FSP.2 and ADV_TDS.1 and AGD_OPE.1 and AGD_PRE.1

Table 5 Requirement Dependencies

The TOE relies on the environment to provide reliable time stamps per OE.TIME. As such, the missing dependency identified in the table above is satisfied with an objective for the operational environment of the TOE.

8.5 TOE Summary Specification Rationale

Each subsection in Section 6, the TOE Summary Specification, describes a security function of the TOE. Each description is followed with rationale that indicates which requirements are satisfied by aspects of the corresponding security function. The set of security functions work together to satisfy all of the security functions and assurance requirements. Furthermore, all of the security functions are necessary in order for the TSF to provide the required security functionality.

This Section in conjunction with Section 6, the TOE Summary Specification, provides evidence that the security functions are suitable to meet the TOE security requirements. The collection of security functions work together to provide all of the security requirements. The security functions described in the TOE summary specification are all necessary for the required security functionality in the TSF. **Table 6 Security Functions vs. Requirements Mapping** demonstrates the relationship between security requirements and security functions.

	Database Discovery and Scanning	Security audit	Identification and authentication	Security management
ADP_RDR_EX.1	X			
ADP_SCN_EX.1	X			
ADP_PRT_EX.1	X			
FAU_GEN.1		X		
FIA_ATD.1			X	
FIA_UID.1			X	
FMT_MOF.1				X
FMT_SMF.1				X
FMT_SMR.1				X

Table 6 Security Functions vs. Requirements Mapping