AppDetective Pro v5.8.0
Security Target

Version 1.0
March 29, 2011

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 Dr.
Columbia, MD 21046

1. SECURITY TARGET INTRODUCTION 4

1.1 SECURITY TARGET, TOE AND CC IDENTIFICATION 4
1.2 CONFORMANCE CLAIMS 4
1.3 CONVENTIONS 5

2. TOE DESCRIPTION 6

2.1 TOE OVERVIEW 6
2.2 TOE ARCHITECTURE 7
2.2.1 Physical Boundary 8
2.2.2 Logical Boundary 9
2.3 TOE DOCUMENTATION 11

3. SECURITY ENVIRONMENT 12

3.1 ORGANIZATIONAL POLICIES 12
3.2 THREATS 12
3.3 ASSUMPTIONS 12

4. SECURITY OBJECTIVES 14

4.1 SECURITY OBJECTIVES FOR THE TOE 14
4.2 SECURITY OBJECTIVES FOR THE IT ENVIRONMENT 14
4.3 SECURITY OBJECTIVES FOR THE ENVIRONMENT 14

5. IT SECURITY REQUIREMENTS 16

5.1 TOE SECURITY FUNCTIONAL REQUIREMENTS 16
5.1.1 DB Scan (EXP) (ADP) 16
5.1.2 Security audit (FAU) 17
5.1.3 Security management (FMT) 17
5.2 IT ENVIRONMENT SECURITY FUNCTIONAL REQUIREMENTS 17
5.2.1 DB Scan (EXP) (ADP) 18
5.2.2 Security audit (FAU) 18
5.2.3 Identification and authentication (FIA) 18
5.2.4 Security management (FMT) 18
5.2.5 Protection of the TSF (FPT) 19
5.3 TOE SECURITY ASSURANCE REQUIREMENTS 19
5.3.1 Configuration management (ACM) 19
5.3.2 Delivery and operation (ADO) 20
5.3.3 Development (ADV) 20
5.3.4 Guidance documents (AGD) 21
5.3.5 Life cycle support (ALC) 22
5.3.6 Tests (ATE) 22
5.3.7 Vulnerability assessment (AVA) 23

6. TOE SUMMARY SPECIFICATION 25

6.1 TOE SECURITY FUNCTIONS 25
6.1.1 Database Discovery and Scanning 25
6.1.2 Security audit 26
6.1.3 Security management 27
6.2 TOE SECURITY ASSURANCE MEASURES 28
6.2.1 Configuration management 28
6.2.2 Delivery and operation 29
6.2.3 Development 29
6.2.4 Guidance documents 29
6.2.5 Life cycle support 30
6.2.6 Tests 30

6.2.7 Vulnerability assessment *30*

7. PROTECTION PROFILE CLAIMS **32**

8. RATIONALE **33**

8.1 SECURITY OBJECTIVES RATIONALE 33
8.1.1 Security Objectives Rationale for the TOE and Environment *33*
8.2 SECURITY REQUIREMENTS RATIONALE 36
8.2.1 Security Functional Requirements Rationale *36*
8.3 SECURITY ASSURANCE REQUIREMENTS RATIONALE 38
8.4 STRENGTH OF FUNCTIONS RATIONALE 39
8.5 REQUIREMENT DEPENDENCY RATIONALE 39
8.6 EXPLICITLY STATED REQUIREMENTS RATIONALE 40
8.7 TOE SUMMARY SPECIFICATION RATIONALE 40
8.8 PP CLAIMS RATIONALE 41

LIST OF TABLES

Table 1 TOE Security Functional Components 16
Table 2 IT Environment Security Functional Components 17
Table 3 EAL 2 augmented with ALC_FLR.2 and AVA_MSU.1 Assurance Components 19
Table 4 Environment to Objective Correspondence 33
Table 5 Objective to Requirement Correspondence 36
Table 6 Security Functions vs. Requirements Mapping 40

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 AppDetective Pro v5.8 provided by Application Security, Inc. The TOE enables enterprise IT security personnel to identify and manage database vulnerabilities. With the reporting capabilities and the policy editor functionality, the TOE provides a GUI interface to manage security risks and extend corporate security policies at the database level.

The Security Target contains the following additional sections:

- TOE Description (Section 2)
- Security Environment (Section 3)
- Security Objectives (Section 4)
- IT Security Requirements (Section 5)
- TOE Summary Specification (Section 6)
- Protection Profile Claims (Section 7)
- Rationale (Section 8)

1.1 Security Target, TOE and CC Identification

ST Title –AppDetective Pro v5.8.0 Security Target

ST Version – Version 1.0

ST Date – March 29, 2011

TOE Identification –AppDetective Pro v5.8.0 (internal version 5.8.7729.0)[1]

TOE Developer – Application Security, Inc.

Evaluation Sponsor – Application Security, Inc.

CC Identification – Common Criteria for Information Technology Security Evaluation, Version 2.3, August 2005

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 2.3, August 2005.
  - Part 2 Extended
- Common Criteria for Information Technology Security Evaluation Part 3: Security Assurance Requirements, Version 2.3, August 2005.
  - Part 3 Conformant
  - Assurance Level: EAL 2 augmented with ALC_FLR.2 and AVA_MSU.1
  - Strength of Function Claim: SOF-basic

[1] This version number can be identified by clicking on "Help->About AppDetective Pro" on the main administrator graphical user interface.

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 …").
    - Explicit: allows the specification of a new class or family of components to be created to address TOE-specific SFRs that are not readily drawn from Part 2 of the CC. This ST contains explicitly Stated Security Functional Requirements. Further, an explicitly stated requirement has (EXP) appended to its name and '_EX' appended to its identifier, before the number, to denote that it has been explicitly stated.
- Other sections of the ST – Other sections of the ST use bolding to highlight text of special interest, such as captions.

2. TOE Description

The Target of Evaluation (TOE) is AppDetective Pro™ V5.8.0, hereafter referred to as AppDetective Pro.

AppDetective Pro 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. AppDetective Pro helps to identify vulnerable databases residing within the network by scanning for potential security vulnerabilities within those databases. Administrators can then take appropriate remedial actions.

AppDetective Pro is one of three related products from Application Security, Inc.; the others being DbProtect AppDetective and DbProtect AppRadar. DbProtect AppDetective incorporates the same database scanning engine as AppDetective Pro, but adds user identification and security management functions to support multiple users in a distributed environment. It also includes the same Policy Editor interface and could be used to manage AppDetective Pro policies if both products are installed on the same host. DbProtect AppRadar actively monitors databases to detect intrusion attempts and generates alarms as appropriate.

AppDetective Pro is an application designed to run in the context of a commercial operating system. Note that while the product comes with the 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.

2.1 TOE Overview

The TOE is a software application that runs in the context of a commercial operating system. AppDetective Pro discovers database applications within an organization's infrastructure and scans them for potential vulnerabilities. AppDetective Pro utilizes a library of known vulnerabilities and misconfiguration signatures. AppDetective Pro includes modules for 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 Pro can generate fix scripts, customized based on scan results, which the administrator[2] can review and apply to address identified vulnerabilities. However, the capabilities of fix scripts have not been assessed as part of the evaluation.

AppDetective Pro 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 Pro 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 Pro 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

[2] 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.

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:

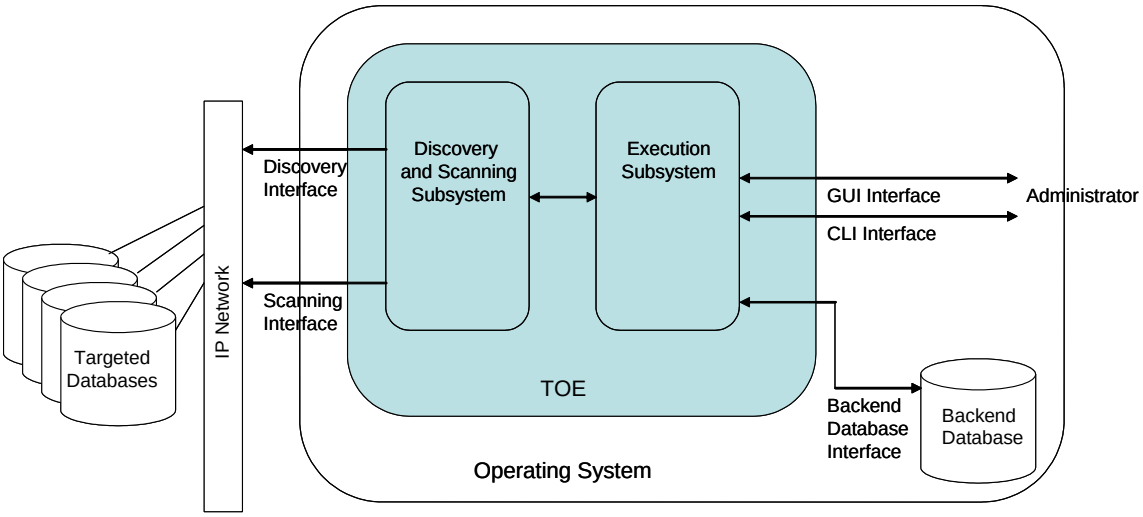
- Access Control—these checks examine the target application for potentially inappropriate or insecure access control or privilege settings on database objects.
- Application Integrity—these checks determine if specific security measures (such as enabling auditing of specific events or encrypting sensitive 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 configuration does not introduce vulnerabilities into the OS (e.g., application processes running with elevated privileges).

AppDetective Pro 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. In addition, the administrator can customize a policy by excluding individual checks (known as exceptions). For example, the administrator can exclude specific accounts from password checks. The effectiveness of policy exclusions was not covered by the evaluation.

2.2 TOE Architecture

The following diagram depicts AppDetective Pro (the TOE) and the environment in which it operates.



The TOE consists of the following subsystems:

- **Execution Subsystem**—provides the Graphical User Interface (GUI) and Command Line Interface (CLI) the administrator uses to manage the TOE and its operations, and the interface to the Backend Database
- **Discovery and Scanning Subsystem**—provides the components that discover applications within the IT environment and performs Pen Tests and Audits against those discovered applications.

Logically, the TOE operates as a single application though it is instantiated as a series of processes utilizing inter-process communication mechanisms provided by the underlying operating system to communicate with one another.

Note that the TOE depends on the underlying operating system to protect its executable and stored data images (e.g., files and registry keys) and its executing environment. The TOE also depends on the environment to provide a secure database store (the Backend Database) for scan results. Note that this database can be on the same machine as the TOE or on a remote machine.

2.2.1 Physical Boundary

The TOE is an application program that operates in the context of a commercial operating system. As such, it utilizes functions of the operating system to execute, store data, and to communicate with database applications on the network as well as for security as indicated above. It implements its own GUI and CLI for its own management and use.

The TOE is designed to operate in the context of the following operating systems: Microsoft Windows XP Professional SP2 or greater; Microsoft Windows Server 2003 Standard Edition; Microsoft Windows Server 2003 Enterprise Edition; Microsoft Windows Server 2003 Enterprise x64 ; Windows Vista (Business, Enterprise and Ultimate editions); and Windows 7 (Professional, Enterprise, and Ultimate editions).

The TOE can be configured to use Microsoft SQL Server 2000 SP4, Microsoft SQL Server 2005, Microsoft SQL Server 2008, Microsoft SQL Express, or MSDE 2000 as its Backed Database (Microsoft Access is not supported). The SQL Server or MSDE 2000 Backend Database can be local or remote to the system hosting the TOE. Note while the product supports the use of MS Access, that database option is excluded from the evaluated configuration.

Additionally, the TOE requires the following components in the IT 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

- SQL Server 2005 Backwards Compatibility (aka Feature Pack for Microsoft SQL Server 2005).

The TOE includes Crystal Reports 9.2.0 to support its report generation and viewing function, WinPcap Pro 4.0.2.1123 to support the Discovery operation, and WodSSH 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, Oracle8i;
- Microsoft SQL Server Versions 2000, 2005, and 2005 Express Edition. MSDE 2000 SP4;
- Lotus Domino 6 and 7;
- Sybase Adaptive Server Enterprise (ASE) 11.9.2, 12.0, 12.5, 15;
- IBM DB2 Version 8.1, IBM DB2 Version 8.2, IBM DB2 Version 9.1, IBM DB2 Version 9.5, IBM DB2 Version 7, 8 and 9 on z/OS and OS/390; and,
- MySQL 4.0, 4.1, 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 IT 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 Boundary

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

2.2.2.1 Evaluated Security Functions

The evaluation addresses the following security functions provided by AppDetective Pro:

- Database Discovery and Scanning
- Security Audit
- Security Management

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 (Crystal Reports, 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. The TOE includes Crystal Reports for generating reports and viewing reports generated in the Crystal Reports format.

Security Audit

The TOE has the ability to generate audit records for the following TOE security-relevant events: creation and modification of scan policies; initiation of Discovery, Pen Test and Audit operations; scheduling of Discovery, Pen Test and Audit operations; creation and review of reports; and generation of fix scripts. The TOE records within each audit record at least the following information: date; time; event type; result; user ID of the user.

Security Management

The TOE provides a graphical user interface and a command line interface for the managing the TOE's security functions and the TOE data. The notion of authorized administrator role, which has full control and privileges to manage the TOE and its security functions, is realized as any user that the IT environment allows to invoke the TOE application.

2.2.2.2 Functions Provided by the IT Environment

The TOE relies on the IT 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 IT environment (i.e., not through the TOE's own interfaces);
- Provide a reliable time stamp for use in audit records and scan results;
- Identify and authenticate authorized administrators and restrict the ability to manage and operate the TOE to authorized administrative users;
- Provide a means for authorized administrators to review the audit records in the audit trail; and,
- Provide encryption services used to encrypt database credentials.

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 TOE 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.

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. In addition, the effectiveness of policy exclusions has not been assessed.

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.

2.3 TOE Documentation

Application Security, Inc. offers a series of documents that describe the installation of AppDetective Pro as well as guidance for subsequent use and administration of the applicable security features (see section 6.2 for details).

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	The authorized administrator of the TOE shall be accountable for using the TOE management functions.[3]
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.[4]
P.MANAGE	The TOE shall provide management functions, which allow the authorized administrators to effectively manage the TOE.[5]

3.2 Threats

T.FACCNT	Unauthorized attempts to access TOE data or security functions may go undetected.
T.LOSSOF	An unauthorized user may attempt to remove or destroy data collected by the TOE.
T.SCNCFG	Improper security configuration settings may exist in the IT System the TOE monitors.
T.SCNVUL	Vulnerabilities may exist in the 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.

[3] Refer to *NIST Special Publication 800-53, Revision 3 Security Controls for Federal Information Systems and Organizations* Audit and Accountability Controls AU-1 and AU-2.
[4] Refer to *NIST Special Publication 800-53, Revision 3 Security Controls for Federal Information Systems and Organizations* Information System Monitoring SI-4.
[5] Refer to *NIST Special Publication 800-53, Revision 3 Security Controls for Federal Information Systems and Organizations* Access Enforcement control AC-3.

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.
A.NOTRST	The TOE can only be accessed by authorized users.

4. Security Objectives

This section defines the security objectives for the TOE and its supporting 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.AUDITS	The TOE must record audit records for data accesses and use of the 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 analyze configuration and vulnerability information that might be indicative of the potential for a future intrusion of an IT System.
O.PROTECT	The TOE must protect any security credentials it uses for the purpose of accessing IT Systems.
O.REPORT	The TOE must provide a capability to generate reports of analytical results of vulnerability scans it performs against IT Systems.

4.2 Security Objectives for the IT Environment

OE.AUDIT	The IT environment of the TOE can audit attempts to access the TOE's stored executable image and stored data.
OE.IDAUTH	The IT environment of the TOE identifies and authenticates users prior to allowing access to TOE.
OE.REVIEW	The IT environment of the TOE provides the capability to review the audit records.
OE.PROTECT	The IT environment of the TOE must protect the TOE from unauthorized modifications and access to stored data.
OE.TIME	The IT Environment will provide reliable timestamps to the TOE.

4.3 Security Objectives for the Environment

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.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.

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 explicitly stated 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 TOE Security Functional Requirements

The following table describes the SFRs that are satisfied by AppDetective Pro.

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.1a: Audit data generation
FMT: Security management	FMT_SMF.1: Specification of Management Functions

Table 1 TOE Security Functional Components

5.1.1 DB Scan (EXP) (ADP)

5.1.1.1 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.

5.1.1.2 DB Scan Data Collection and Analysis (EXP) (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.

5.1.1.3 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.

5.1.2 Security Audit (FAU)

5.1.2.1 Audit Data Generation (FAU_GEN.1a)

FAU_GEN.1a.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) **[**
creation and modification of the AppDetective scan policies,
addition or removal of target database applications,
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, and
generation of fix scripts based on DB scan data]

FAU_GEN.1a.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, 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**].

5.1.3 Security Management (FMT)

5.1.3.1 Specification of Management Functions (FMT_SMF.1)

FMT_SMF.1.1 The TSF shall be capable of performing the following security management functions: [
creation and modification of the AppDetective policies,
addition or removal of target database applications,
initiation of Discovery, Pen Test, or Audit scans,
scheduling Discovery, Pen test, or Audit scans, and
creation and review of DB Scan reports based on DB scan data].

5.2 IT Environment Security Functional Requirements

The following table describes the SFRs that are to be satisfied by the IT environment of AppDetective Pro.

Requirement Class	Requirement Component
ADP: DB Scan (EXP)	ADP_STG_EX.1: DB Scan Data Storage (EXP)
FAU: Security audit	FAU_GEN.1b: Audit data generation
	FAU_SAR.1: Audit review
FIA: Identification and authentication	FIA_ATD.1: User attribute definition
	FIA_UAU.2: User authentication before any action
	FIA_UID.2: User identification before any action
FMT: Security management	FMT_MTD.1: Management of TSF data
	FMT_MOF.1: Management of security functions behavior
	FMT_SMR.1: Security roles
FPT: Protection of the TSF	FPT_RVM.1: Non-bypassability of the TSP
	FPT_SEP.1: TSF domain separation
	FPT_STM.1: Reliable time stamps

Table 2 IT Environment Security Functional Components

5.2.1 DB Scan (EXP) (ADP)

5.2.1.1 DB Scan Data Storage (EXP) (ADP_STG_EX.1)

ADP_STG_EX.1.1 The IT environment shall protect the stored DB Scan data and audit data from unauthorized deletion.

ADP_STG_EX.1.2 The IT environment shall protect the stored DB Scan data and audit data from modification.

5.2.2 Security Audit (FAU)

5.2.2.1 Audit Data Generation (FAU_GEN.1b)

FAU_GEN.1b.1 The **IT environment** shall be able to generate an audit record of the following auditable events:
a) Start-up and shutdown of the audit functions **of the IT environment**;
b) All auditable events for the [***not specified***] level of audit; and
c) **[attempts to access the TOE's stored executable image and stored data]**

FAU_GEN.1b.2 The **IT environment** shall record within each audit record at least the following information:
a) Date and time of the event, type of event, subject identity, 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**].

5.2.2.2 Audit Review (FAU_SAR.1)

FAU_SAR.1.1 The **IT environment** shall provide [**authorized administrator**] with the capability to read [**all audit data**] from the audit records.

FAU_SAR.1.2 The **IT environment** shall provide the audit records in a manner suitable for the user to interpret the information.

5.2.3 Identification and Authentication (FIA)

5.2.3.1 User Attribute Definition (FIA_ATD.1)

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

5.2.3.2 User Authentication Before any Action (FIA_UAU.2)

FIA_UAU.2.1 The **IT environment** shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

5.2.3.3 User Identification Before any Action (FIA_UID.2)

FIA_UID.2.1 The **IT environment** shall require each user to identify itself 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 Behavior (FMT_MOF.1)

FMT_MOF.1.1 The **IT environment** shall restrict the ability to [***modify the behavior of***] the functions [**all TOE security functions**] to [**the authorized administrator**].

5.2.4.2 Management of TSF Data (FMT_MTD.1)

FMT_MTD.1.1 The **IT environment** shall restrict the ability to [***query, modify, or delete***] the [**all TOE data**] to [**authorized administrator**].

5.2.4.3 Security Roles (FMT_SMR.1)

FMT_SMR.1.1 The **IT environment** shall maintain the roles [**authorized administrator**].
FMT_SMR.1.2 The **IT environment** shall be able to associate users with roles.

5.2.5 Protection of the TSF (FPT)

5.2.5.1 Non-Bypassability of the TSP (FPT_RVM.1)

FPT_RVM.1.1 The **IT environment** shall ensure that TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed.

5.2.5.2 TSF Domain Separation (FPT_SEP.1)

FPT_SEP.1.1 The **IT environment** shall maintain a security domain for its own execution **and that of the TOE** that protects it from interference and tampering by untrusted subjects.
FPT_SEP.1.2 The ~~TSF~~ **IT environment** shall enforce separation between the security domains of subjects in the TSC.

5.2.5.3 Reliable Time Stamps (FPT_STM.1)

FPT_STM.1.1 The **IT environment** shall be able to provide reliable time stamps for its own use **as well as that of the TOE**.

5.3 TOE Security Assurance Requirements

The security assurance requirements for the TOE are the EAL 2 augmented with ALC_FLR.2 and AVA_MSU.1 components as specified in Part 3 of the Common Criteria.

Requirement Class	Requirement Component
ACM: Configuration management	ACM_CAP.2: Configuration items
ADO: Delivery and operation	ADO_DEL.1: Delivery procedures
	ADO_IGS.1: Installation, generation, and start-up procedures
ADV: Development	ADV_FSP.1: Informal functional specification
	ADV_HLD.1: Descriptive high-level design
	ADV_RCR.1: Informal correspondence demonstration
AGD: Guidance documents	AGD_ADM.1: Administrator guidance
	AGD_USR.1: User guidance
ALC: Life cycle support	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_MSU.1: Examination of guidance
	AVA_SOF.1: Strength of TOE security function evaluation
	AVA_VLA.1: Developer vulnerability analysis

Table 3 EAL 2 augmented with ALC_FLR.2 and AVA_MSU.1 Assurance Components

5.3.1 Configuration Management (ACM)

5.3.1.1 Configuration Items (ACM_CAP.2)

ACM_CAP.2.1d The developer shall provide a reference for the TOE.

ACM_CAP.2.2d The developer shall use a CM system.
ACM_CAP.2.3d The developer shall provide CM documentation.
ACM_CAP.2.1c The reference for the TOE shall be unique to each version of the TOE.
ACM_CAP.2.2c The TOE shall be labeled with its reference.
ACM_CAP.2.3c The CM documentation shall include a configuration list.
ACM_CAP.2.4c The configuration list shall uniquely identify all configuration items that comprise the TOE.
ACM_CAP.2.5c The configuration list shall describe the configuration items that comprise the TOE.
ACM_CAP.2.6c The CM documentation shall describe the method used to uniquely identify the configuration items that comprise the TOE.
ACM_CAP.2.7c The CM system shall uniquely identify all configuration items that comprise the TOE.
ACM_CAP.2.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.2 Delivery and Operation (ADO)

5.3.2.1 Delivery Procedures (ADO_DEL.1)

ADO_DEL.1.1d The developer shall document procedures for delivery of the TOE or parts of it to the user.
ADO_DEL.1.2d The developer shall use the delivery procedures.
ADO_DEL.1.1c The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to a user's site.
ADO_DEL.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.2.2 Installation, Generation, and Start-Up Procedures (ADO_IGS.1)

ADO_IGS.1.1d The developer shall document procedures necessary for the secure installation, generation, and start-up of the TOE.
ADO_IGS.1.1c The installation, generation and start-up documentation shall describe all the steps necessary for secure installation, generation and start-up of the TOE.
ADO_IGS.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADO_IGS.1.2e The evaluator shall determine that the installation, generation, and start-up procedures result in a secure configuration.

5.3.3 Development (ADV)

5.3.3.1 Informal Functional Specification (ADV_FSP.1)

ADV_FSP.1.1d The developer shall provide a functional specification.
ADV_FSP.1.1c The functional specification shall describe the TSF and its external interfaces using an informal style.
ADV_FSP.1.2c The functional specification shall be internally consistent.
ADV_FSP.1.3c The functional specification shall describe the purpose and method of use of all external TSF interfaces, providing details of effects, exceptions and error messages, as appropriate.
ADV_FSP.1.4c The functional specification shall completely represent the TSF.
ADV_FSP.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_FSP.1.2e The evaluator shall determine that the functional specification is an accurate and complete instantiation of the TOE security functional requirements.

5.3.3.2 Descriptive High-Level Design (ADV_HLD.1)

ADV_HLD.1.1d The developer shall provide the high-level design of the TSF.
ADV_HLD.1.1c The presentation of the high-level design shall be informal.
ADV_HLD.1.2c The high-level design shall be internally consistent.
ADV_HLD.1.3c The high-level design shall describe the structure of the TSF in terms of subsystems.

ADV_HLD.1.4c The high-level design shall describe the security functionality provided by each subsystem of the TSF.
ADV_HLD.1.5c The high-level design shall identify any underlying hardware, firmware, and/or software required by the TSF with a presentation of the functions provided by the supporting protection mechanisms implemented in that hardware, firmware, or software.
ADV_HLD.1.6c The high-level design shall identify all interfaces to the subsystems of the TSF.
ADV_HLD.1.7c The high-level design shall identify which of the interfaces to the subsystems of the TSF are externally visible.
ADV_HLD.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_HLD.1.2e The evaluator shall determine that the high-level design is an accurate and complete instantiation of the TOE security functional requirements.

5.3.3.3 Informal Correspondence Demonstration (ADV_RCR.1)

ADV_RCR.1.1d The developer shall provide an analysis of correspondence between all adjacent pairs of TSF representations that are provided.
ADV_RCR.1.1c For each adjacent pair of provided TSF representations, the analysis shall demonstrate that all relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.
ADV_RCR.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.4 Guidance Documents (AGD)

5.3.4.1 Administrator Guidance (AGD_ADM.1)

AGD_ADM.1.1d The developer shall provide administrator guidance addressed to system administrative personnel.
AGD_ADM.1.1c The administrator guidance shall describe the administrative functions and interfaces available to the administrator of the TOE.
AGD_ADM.1.2c The administrator guidance shall describe how to administer the TOE in a secure manner.
AGD_ADM.1.3c The administrator guidance shall contain warnings about functions and privileges that should be controlled in a secure processing environment.
AGD_ADM.1.4c The administrator guidance shall describe all assumptions regarding user behaviour that are relevant to secure operation of the TOE.
AGD_ADM.1.5c The administrator guidance shall describe all security parameters under the control of the administrator, indicating secure values as appropriate.
AGD_ADM.1.6c The administrator guidance shall describe each type of security-relevant event relative to the administrative functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.
AGD_ADM.1.7c The administrator guidance shall be consistent with all other documentation supplied for evaluation.
AGD_ADM.1.8c The administrator guidance shall describe all security requirements for the IT environment that are relevant to the administrator.
AGD_ADM.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.4.2 User Guidance (AGD_USR.1)

AGD_USR.1.1d The developer shall provide user guidance.
AGD_USR.1.1c The user guidance shall describe the functions and interfaces available to the non-administrative users of the TOE.
AGD_USR.1.2c The user guidance shall describe the use of user-accessible security functions provided by the TOE.
AGD_USR.1.3c The user guidance shall contain warnings about user-accessible functions and privileges that should be controlled in a secure processing environment.

AGD_USR.1.4c The user guidance shall clearly present all user responsibilities necessary for secure operation of the TOE, including those related to assumptions regarding user behaviour found in the statement of TOE security environment.
AGD_USR.1.5c The user guidance shall be consistent with all other documentation supplied for evaluation.
AGD_USR.1.6c The user guidance shall describe all security requirements for the IT environment that are relevant to the user.
AGD_USR.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.5 Life Cycle Support (ALC)

5.3.5.1 Flaw Reporting Procedures (ALC_FLR.2)

ALC_FLR.2.1d The developer shall document the flaw remediation procedures.
ALC_FLR.2.2d The developer shall establish a procedure for accepting and acting upon user reports of security flaws and requests for corrections to those flaws.
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 procedures for processing reported security flaws shall ensure that any reported flaws are corrected and the correction issued to TOE users.
ALC_FLR.2.6c 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.2.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.6 Tests (ATE)

5.3.6.1 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 identified in the test documentation and the TSF as described 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.

5.3.6.2 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, test procedure descriptions, expected test results and actual test results.
ATE_FUN.1.2c The test plans shall identify the security functions to be tested and describe the goal of the tests to be performed.
ATE_FUN.1.3c The test procedure descriptions shall identify the tests to be performed and describe the scenarios for testing each security function. These scenarios shall include any ordering dependencies on the results of other tests.
ATE_FUN.1.4c The expected test results shall show the anticipated outputs from a successful execution of the tests.
ATE_FUN.1.5c The test results from the developer execution of the tests shall demonstrate that each tested security function behaved as specified.

ATE_FUN.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.6.3 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 test a subset of the TSF as appropriate to confirm that the TOE operates as specified.
ATE_IND.2.3e The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.

5.3.7 Vulnerability Assessment (AVA)

5.3.7.1 Examination of Guidance (AVA_MSU.1)

AVA_MSU.1.1d The developer shall provide guidance documentation.
AVA_MSU.1.1c The guidance documentation 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.
AVA_MSU.1.2c The guidance documentation shall be complete, clear, consistent and reasonable.
AVA_MSU.1.3c The guidance documentation shall list all assumptions about the intended environment.
AVA_MSU.1.4c The guidance documentation shall list all requirements for external security measures (including external procedural, physical and personnel controls).
AVA_MSU.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
AVA_MSU.1.2e The evaluator shall repeat all configuration and installation procedures to confirm that the TOE can be configured and used securely using only the supplied guidance documentation.
AVA_MSU.1.3e The evaluator shall determine that the use of the guidance documentation allows all insecure states to be detected.

5.3.7.2 Strength of TOE Security Function Evaluation (AVA_SOF.1)

AVA_SOF.1.1d The developer shall perform a strength of TOE security function analysis for each mechanism identified in the ST as having a strength of TOE security function claim.
AVA_SOF.1.1c For each mechanism with a strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the minimum strength level defined in the PP/ST.
AVA_SOF.1.2c For each mechanism with a specific strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the specific strength of function metric defined in the PP/ST.
AVA_SOF.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
AVA_SOF.1.2e The evaluator shall confirm that the strength claims are correct.

5.3.7.3 Developer Vulnerability Analysis (AVA_VLA.1)

AVA_VLA.1.1d The developer shall perform a vulnerability analysis.
AVA_VLA.1.2d The developer shall provide vulnerability analysis documentation.
AVA_VLA.1.1c The vulnerability analysis documentation shall describe the analysis of the TOE deliverables performed to search for obvious ways in which a user can violate the TSP.
AVA_VLA.1.2c The vulnerability analysis documentation shall describe the disposition of obvious vulnerabilities.
AVA_VLA.1.3c The vulnerability analysis documentation shall show, for all identified vulnerabilities, that the vulnerability cannot be exploited in the intended environment for the TOE.

AVA_VLA.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_VLA.1.2e The evaluator shall conduct penetration testing, building on the developer vulnerability analysis, to ensure obvious vulnerabilities have been addressed.

6. TOE Summary Specification

This chapter describes the security functions and their associated assurance measures.

6.1 TOE Security Functions

6.1.1 Database Discovery and Scanning

The TOE provides a graphical user interface (GUI) and a Command Line Interface (CLI) 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 TOE is capable of performing various checks on targeted 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 IT 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 IP address range or for an entire network. A Discovery locates database applications on the network, identifies the database application's IP 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 the AppDetective Pro. The reports present the scan results in a human readable format and can include all the information from those results. AppDetective Pro 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. If a user interactively launches a scan, the user will be prompted for database credentials that are used and then discarded within the memory of the TOE (these credentials are not persistently stored by the TOE). However, for scans scheduled using the Job Scheduler (see Section 6.1.3), applicable security credentials can be configured into the TOE by the user and the TOE will protect those credentials using the Windows Data Protection API (DPAPI) provided by its host operating system to encrypt them and will store them in its backend database until they are no longer needed (i.e., until a scheduled job executes or a recurring job is deleted). Additionally, the TOE calls upon the Windows DPAPI to encrypt the credentials used to access its backend database and stores them in the Windows 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 user 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.

6.1.2 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 to protect and store the audit records, provide the ability to review the audit records, and to provide a reliable timestamp. The TOE is able to generate audit records for the following TOE security relevant events:

- Creation and modification of the AppDetective scan policies
  - Risk Level Modification: The authorized administrator can modify the system setting for the risk level of a check. Each check is associated with a risk level and the checks performed can be limited by changing the risk level.

    - Exception Creation/Edit/Deletion: Within the policy editor, the authorized administrator may create/edit/delete an exception. Specific scan checks can be identified as exceptions that will not be checked and reported.
    - Export/Import/Purge Policy: Importing, exporting or purging the policies for the future jobs.
    - User-Defined Check Creation/Edit/Deletion: The TOE allows the authorized administrator to create user-defined checks; the customized SQL code enhances the existing Policies.[6]
- Addition or removal of target database applications
    - Application Addition/Removal: Adding or removing the applications to the audits/pen tests.
    - Session Creation: Creates new sessions to communicate with the targeted IT systems.
    - Sessions Merged: A Session is a logical grouping of applications and the Pen Tests/Audits run against a group of application. Sessions can be merged.
    - Export/Import/Purge Session: A Session is a logical grouping of applications and the Pen Tests/Audits run against them. Sessions can be exported, imported, or purged.
- Initiation of Discovery, Pen Test, or Audit scans
    - Discovery Performed (include scheduled): Performing the Discovery job.
    - Pen Test Performed (include scheduled): The performed pen tests.
    - Audit Performed (include scheduled): Auditing the targeted IT systems.
- Scheduling Discovery, Pen test, or Audit scans
    - Scheduled Job Addition/Deletion: The authorized administrator may add or delete scheduled jobs.
- Creation and review of DB Scan reports based on DB scan data
    - Report Creation: Creating the reports to be reviewed by the authorized administrators.
    - Vulnerability Suppression: A found vulnerability can be suppressed from a Report. The user can suppress a report by using the Vulnerability Manager, or with in the Main View of the Vulnerabilities Tab.
- Generation of fix scripts based on DB scan data
    - Fix Script Creation: The Fix Scripts utility generates SQL scripts designed to correct misconfigurations and address vulnerabilities identified by AppDetective during an Audit.

Note that although the TOE provides the ability to generate Fix Scripts and audit that generation, no assessment has been made of the quality of the generated fix scripts.

The TOE provides the capability to enable and disable auditing. These actions are also audited

Each audit record is generated with the date and time queried from the operating system and the type of event per the list above. The subject is obtained from the operating system in the environment since the TOE depends upon the operating system for user identity information. The outcome for each event is always logged as successful on completion.

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.

6.1.3 Security Management

The TOE provides a GUI and a CLI to enable the authorized administrator to manage the TOE security functions. Any user that can access the TOE via the IT environment either through the GUI or the CLI is considered an authorized administrator role. The TOE relies on the underlying Operating System to provide identification and authentication and maintain the authorized administrator role. The TOE allows the authorized administrator to manage the following tasks (note that specific TOE functions are mapped to more general security management function classes):

[6] Given the dynamic and user-defined nature of this capability, it has not been subject to analysis and testing within the scope of the evaluation.

- Creation and modification of the AppDetective scan policies
  - **Policy:** Sets of security checks used when AppDetective performs Pen Tests and Audits. AppDetective contains several built-in Pen Test and Audit Policies. In addition, new Policies can be created and modified. The built-in scan policies cannot be deleted.
  - **User-Defined Checks:** Define the MS-SQL, DB2, Sybase and Oracle checks to supplement the built-in AppDetective security checks.[7]
- Addition or removal of target database applications
  - **Edit Menu Tasks:** Add applications for AppDetective to Pen Test or Audit, organize the AppDetective export/ purge data.
  - **Session:** Specifies the types of applications and range of ports on the targeted network for the Pen Test and Audit. The Session is a prerequisite for most AppDetective tasks.
- Initiation of Discovery, Pen Test, or Audit scans
  - **Discovery:** Discovery function locates network applications (and identifies their IP addresses), as well as the ports used to provide network services. The Pen Tests and Audits are executed against discovered applications and ports.
  - **Pen Tests and Audits:** Find internal and external vulnerabilities in the discovered applications (according to the Policy selected).
- Scheduling Discovery, Pen test, or Audit scans
  - **Job Scheduler:** Schedule the date and time to run an AppDetective task, such as a Pen Test or Audit. Note that in addition to utilizing host operating system features to schedule jobs, the Job Scheduler allows the user to configure specific e-mail addresses. When a scheduled job completes, the TOE will use e-mail services provided by its environment to send the scan results to the configured users.
- Creation and review of DB Scan reports based on DB scan data
  - **Reports:** Communicate vulnerabilities discovered by AppDetective (and actions taken) to all levels of the organization.
  - **Vulnerability Manager:** Manage security vulnerabilities found in a Session and apply filters to help assess the status of various application vulnerabilities.

Note that although the TOE provides the ability to generate Fix Scripts and audit that generation, no assessment has been made of the quality of the generated fix scripts.

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

- FMT_SMF.1: The TOE is able to perform management functions (management of AppDetective tasks and data).

6.2 TOE Security Assurance Measures

6.2.1 Configuration Management

The configuration management measures applied by Application Security ensure that configuration items are uniquely identified, and that documented procedures are used to control and track changes that are made to the TOE. Application Security performs configuration management on the TOE implementation representation, design, tests, user and administrator guidance, and the CM documentation.

These activities are documented in:

[7] Given the dynamic and user-defined nature of this capability, it has not been subject to analysis and testing within the scope of the evaluation.

- DbProtect AppRadar, DbProtect AppDetective, AppDetective Pro Configuration Management Plan

The Configuration management assurance measure satisfies the following EAL 2 augmented with ALC_FLR.1 and AVA_MSU.1 assurance requirements:

- ACM_CAP.2

6.2.2 Delivery and Operation

Application Security provides delivery documentation and procedures to identify the TOE, secure the TOE during delivery, and provide necessary installation and generation instructions. Application Security's delivery procedures describe all applicable procedures to be used to prevent in appropriate access to the TOE. Application Security also provides documentation that describes the steps necessary to install AppDetective in accordance with the evaluated configuration.

These activities are documented in:

- AppDetective Pro 5.8 Installation and User's Guide
- DbProtect AppRadar DbProtect AppDetective AppDetective Pro Delivery Procedures
- Installing and Using the Common Criteria Configuration of AppDetective Pro v5.8

The Delivery and operation assurance measure satisfies the following EAL 2 augmented with ALC_FLR.2 and AVA_MSU.1 assurance requirements:

- ADO_DEL.1
- ADO_IGS.1

6.2.3 Development

Application Security has documents describing all facets of the design of the TOE. These documents serve to describe the security functions of the TOE, its interfaces both external and between subsystems, the architecture of the TOE (in terms of subsystems), and correspondence between the available design abstractions (including the ST).

These activities are documented in:

- Application Security AppDetective Pro V5.8.0 Functional Specification and High Level Design Document for Common Criteria Evaluation

The Development assurance measure satisfies the following EAL 2 augmented with ALC_FLR.2 and AVA_MSU.1 assurance requirements:

- ADV_FSP.1
- ADV_HLD.1
- ADV_RCR.1

6.2.4 Guidance Documents

Application Security provides administrator and user guidance on how to utilize the TOE security functions and warnings to administrators and users about actions that can compromise the security of the TOE.

These activities are documented in:

- AppDetective Pro 5.8 Installation and User's Guide
- Installing and Using the Common Criteria Configuration of AppDetective Pro v5.8

The Guidance documents assurance measure satisfies the following EAL 2 augmented with ALC_FLR.2 and AVA_MSU.1 assurance requirements:

- AGD_ADM.1
- AGD_USR.1

6.2.5 Life Cycle Support

Application Security has procedures that define the process for accepting and acting upon user reports of security flaws. These procedures describe the acceptance criteria for security flaws and how all security flaws and the status of fixes for each security flaw are tracked.

These activities are documented in:

- DbProtect AppRadar, DbProtect AppDetective, AppDetective Pro Life Cycle Document

The Life cycle support assurance measure satisfies the following EAL 2 augmented with ALC_FLR.2 and AVA_MSU.1 assurance requirements:

- ALC_FLR.2

6.2.6 Tests

The test documents describe the overall test plan, testing procedures, the tests themselves, including expected and actual results. In addition, these documents describe how the functional specification has been appropriately tested.

These activities are documented in:

- AppDetective Pro 5.8.0 Test Plan & Vulnerability Assessment
- Application Security AppDetective Test Procedures
- Application Security AppDetective Test Results

The Tests assurance measure satisfies the following EAL 2 augmented with ALC_FLR.2 and AVA_MSU.1 assurance requirements:

- ATE_COV.1
- ATE_FUN.1
- ATE_IND.2

6.2.7 Vulnerability Assessment

The TOE administrator and user guidance documents describe the operation of AppDetective and how to maintain a secure state. These guides also describe all necessary operating assumptions and security requirements outside the scope of control of the TOE. They have been developed to serve as complete, clear, consistent, and reasonable administrator and user references.

Application Security has not conducted a strength of function analysis since the TOE includes no permutational or probabilistic security mechanisms. Otherwise the minimum strength of function claim, SOF-basic, would have served as the minimum standard of acceptability for applicable mechanisms.

Application Security performs regular vulnerability analyses of the entire TOE (including documentation) to identify obvious weaknesses that can be exploited in the TOE.

These activities are documented in:

- AppDetective Pro 5.8.0 Test Plan & Vulnerability Assessment

The Vulnerability assessment assurance measure satisfies the following EAL 2 augmented with ALC_FLR.2 and AVA_MSU.1 assurance requirements:

- AVA_MSU.1

- AVA_SOF.1
- AVA_VLA.1

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;
- Strength of Functions;
- Requirement Dependencies;
- TOE Summary Specification; and,
- PP Claims.

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 Environment

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

	P.ACCACT	P.DETECT	P.MANAGE	T.FACCNT	T.LOSSOF	T.SCNCFG	T.SCNVUL	A.ACCESS	A.DYNMIC	A.MANAGE	A.NOEVIL	A.NOTRST
O.AUDITS	X			X								
O.EADMIN			X									
O.IDACTS		X				X	X					
O.PROTECT		X										
O.REPORT						X	X					
OE.AUDIT				X								
OE.IDAUTH	X		X		X							
OE.REVIEW	X			X								
OE.TIME	X											
OE.PROTECT					X							
OE.CREDEN			X								X	X
OE.INSTAL			X								X	
OE.INTROP								X	X			
OE.PERSON			X						X	X		
OE.PHYCAL												X

Table 4 Environment to Objective Correspondence

8.1.1.1 P.ACCACT

The authorized administrator of the TOE shall be accountable for using the TOE management functions.

This Organizational Policy is satisfied by ensuring that:

- O.AUDITS: The O.AUDITS objective implements this policy by requiring auditing of all data accesses and use of TOE functions.
- OE.IDAUTH: The OE.IDAUTH objective supports this policy by ensuring the AppDetective user is uniquely identified and authenticated.
- OE.REVIEW: The OE.REVIEW objective supports this policy by ensuring the IT 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 IT environment.

8.1.1.2 P.DETECT

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.
- O.PROTECT: The O.PROTECT objective serves to ensure that any credentials used in the process of collecting data (e.g., configuration data) is protected so as not to introduce vulnerabilities into those systems being accessed.

8.1.1.3 P.MANAGE

The TOE shall provide management functions, which allow the authorized administrators to effectively manage the TOE.

This Organizational Policy is satisfied by ensuring that:

- O.EADMIN: the O.EADMIN objective ensures there is a set of functions for administrators to use.
- OE.IDAUTH: The OE.IDAUTH objective ensures the identification and authentication of users prior to any TOE function accesses.
- 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.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 IT 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 ensures the IT environment of the TOE provides the capability to review the generated TOE audit records.

8.1.1.5 T.LOSSOF

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

This Threat is satisfied by ensuring that:

- OE.IDAUTH: The OE.IDAUTH objective provides for identification and authentication of users prior to any TOE data access.
- OE.PROTECT: The OE.PROTECT objective ensures that the environment provides a secure environment for the TOE, including in addition to protection of the TOE itself protection for access to the TOE's data.

8.1.1.6 T.SCNCFG

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

This Threat is satisfied by ensuring that:

- 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.
- O.REPORT: The O.REPORT objective supports O.IDACTS by requiring the TOE provide a capability to generate reports of the results of vulnerability scans it performs against IT Systems.

8.1.1.7 T.SCNVUL

Vulnerabilities may exist in the IT System the TOE monitors.

This Threat is satisfied by ensuring that:

- 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.
- O.REPORT: The O.REPORT objective supports O.IDACTS by requiring the TOE provide a capability to generate reports of the results of vulnerability scans it performs against IT Systems.

8.1.1.8 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.

8.1.1.9 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 be managed appropriately.

8.1.1.10 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.

8.1.1.11 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.

8.1.1.12 A.NOTRST

The TOE can only be accessed by authorized users.

This Assumption is satisfied by ensuring that:

- OE.CREDEN: The OE.CREDEN objective supports this assumption by requiring protection of applicable authentication data.
- OE.PHYCAL: The OE.PHYCAL objective provides for physical protection of the TOE to protect against unauthorized access.

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 5** 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.AUDITS	O.EADMIN	O.IDACTS	O.PROTECT	O.REPORT	OE.AUDIT	OE.IDAUTH	OE.REVIEW	OE.PROTECT	OE.TIME
ADP_RDR_EX.1		X			X					
ADP_SCN_EX.1			X							
ADP_PRT_EX.1				X						
FAU_GEN.1a	X									
FMT_SMF.1		X								
ADP_STG_EX.1									X	
FAU_GEN.1b						X				
FAU_SAR.1								X		
FIA_ATD.1							X			
FIA_UAU.2							X			
FIA_UID.2							X			
FMT_MOF.1									X	
FMT_MTD.1									X	
FMT_SMR.1							X			
FPT_RVM.1									X	
FPT_SEP.1									X	
FPT_STM.1										X

Table 5 Objective to Requirement Correspondence

8.2.1.1 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.1a: Security-relevant events must be defined and auditable for the TOE.

8.2.1.2 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 requires specified security management functions.

8.2.1.3 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 TOE is required to collect and store configuration and vulnerability information of an IT System.

8.2.1.4 O.PROTECT

The TOE must protect any security credentials it uses for the purpose of accessing IT Systems.

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.

8.2.1.5 O.REPORT

The TOE must provide a capability to generate reports of the results of vulnerability scans it performs against IT Systems.

This TOE Security Objective is satisfied by ensuring that:

- ADP_RDR_EX.1: The TOE is required to provide a capability to create and review scan reports based on the scan data collected by the TSF and present them in a manner suitable for the user to interpret.

8.2.1.6 OE.AUDIT

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

This IT Environment Security Objective is satisfied by ensuring that:

- FAU_GEN.1b: Attempts to access the TOE executable or data can be audited.

8.2.1.7 OE.IDAUTH

The IT environment of the TOE identifies and authenticates users prior to allowing access to TOE.

This IT Environment Security Objective is satisfied by ensuring that:

- FMT_SMR.1: The IT environment of the TOE must be able to recognize the different administrative and user roles that exist for the TOE.
- FIA_ATD.1: Security attributes of subjects use to enforce the authentication policy of the TOE must be defined.

- FIA_UAU.2: The IT environment of the TOE successfully authenticates the authorized administrator before allowing any access to the TOE.
- FIA_UID.2: The IT environment of the TOE successfully identifies the authorized administrator before allowing any access to the TOE.

8.2.1.8 OE.REVIEW

The IT environment of the TOE provides the capability to review the audit records.

This IT Environment Security Objective is satisfied by ensuring that:

- FAU_SAR.1: The environment of the TOE provides capability to read all audit records generated by the TOE.

8.2.1.9 OE.PROTECT

The IT environment of the TOE must protect the TOE from unauthorized modifications and access to stored data.

This IT Environment Security Objective is satisfied by ensuring that:

- ADP_STG_EX.1: The environment of the TOE protects the stored DB Scan data and audit data from unauthorized deletion and modification.
- FMT_MOF.1: The environment of the TOE protects the ability to modify the behavior of the TOE.
- FMT_MTD.1: The environment of the TOE protects the ability to access (query, modify, or delete) TOE data.
- FPT_RVM.1: The IT environment of the TOE must ensure that the resources and security domain of the TOE are protected so that they cannot be accessed by untrusted users. The IT environment is expected to prevent bypass of TOE mechanisms by ensuring that the data or domains that it instantiates are appropriately protected.
- FPT_SEP.1: The IT environment of the TOE must ensure that the security domain of the TOE is protected so that is cannot be accessed or tampered with by untrusted users. The IT environment is expected to prevent tampering of the TOE by ensuring that its execution domain is not accessible by untrusted users.

8.2.1.10 OE.TIME

The IT Environment will provide reliable timestamps to the TOE.

This IT Environment Security Objective is satisfied by ensuring that:

- FPT_STM.1: The IT environment must provide reliable time stamps.

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, AppDetective is targeted at an environment with good physical access security and competent administrators. Within such environments, 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.

Note that AVA_MSU.1 has been added since it is important that the guidance documentation for the TOE to be complete and to mitigate potential misuse of the security functions. ALC_FLR.2 has been added since it is important to accept reported flaws and to make appropriate corrections to counter confirmed flaws.

8.4 Strength of Functions Rationale

The overall strength of function claim of SOF-basic is believed to be commensurate with the overall assurance claim of EAL 2. The TOE does not provide any applicable permutational and probabilistic mechanism; therefore, the overall strength of function claim of SOF-basic is believed to be sufficient.

8.5 Requirement Dependency Rationale

The following table identifies the dependencies of the requirements in this ST, including the requirements explicitly defined in this ST. As indicated in the table, all of the dependencies are satisfied, except those identified in [bold-red-bracketed] text. It is expected that the environment would further satisfy its own requirements, and the TOE is not dependent upon the manner in which that is accomplished. As such, the applicable dependencies are not specifically identified in this ST.

ST Requirement	CC Dependencies	ST Dependencies
ADP_RDR_EX.1	ADP_SCN_EX.1 (per section 8.6)	ADP_SCN_EX.1
ADP_SCN_EX.1	FPT_STM.1 (per section 8.6)	*FPT_STM.1*
ADP_PRT_EX.1	none	none
FAU_GEN.1a	FPT_STM.1	*FPT_STM.1*
FMT_SMF.1	none	none
ADP_STG_EX.1	none	none
FAU_GEN.1a	FPT_STM.1	*FPT_STM.1*
FAU_SAR.1	FAU_GEN.1	FAU_GEN.1a and FAU_GEN.1b
FIA_ATD.1	none	none
FIA_UAU.2	FIA_UID.1	*FIA_UID.2*
FIA_UID.2	none	none
FMT_MOF.1	FMT_SMR.1	*FMT_SMR.1*
FMT_MTD.1	FMT_SMR.1	*FMT_SMR.1*
FMT_SMR.1	FIA_UID.1	*FIA_UID.2*
FPT_RVM.1	none	none
FPT_SEP.1	none	none
FPT_STM.1	none	none
ACM_CAP.2	none	none
ADO_DEL.1	none	none
ADO_IGS.1	AGD_ADM.1	AGD_ADM.1
ADV_FSP.1	ADV_RCR.1	ADV_RCR.1
ADV_HLD.1	ADV_FSP.1 and ADV_RCR.1	ADV_FSP.1 and ADV_RCR.1
ADV_RCR.1	none	none
AGD_ADM.1	ADV_FSP.1	ADV_FSP.1
AGD_USR.1	ADV_FSP.1	ADV_FSP.1
ALC_FLR.2	none	none
ATE_COV.1	ADV_FSP.1 and ATE_FUN.1	ADV_FSP.1 and ATE_FUN.1
ATE_FUN.1	none	none
ATE_IND.2	ADV_FSP.1 and AGD_ADM.1 and AGD_USR.1 and ATE_FUN.1	ADV_FSP.1 and AGD_ADM.1 and AGD_USR.1 and ATE_FUN.1
AVA_MSU.1	ADO_IGS.1 and ADV_FSP.1 and AGD_ADM.1 and AGD_USR.1	ADO_IGS.1 and ADV_FSP.1 and AGD_ADM.1 and AGD_USR.1
AVA_SOF.1	ADV_FSP.1 and ADV_HLD.1	ADV_FSP.1 and ADV_HLD.1
AVA_VLA.1	ADV_FSP.1 and ADV_HLD.1 and AGD_ADM.1 and AGD_USR.1	ADV_FSP.1 and ADV_HLD.1 and AGD_ADM.1 and AGD_USR.1

8.6 Explicitly Stated Requirements Rationale

A family 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 family of FAU was initially used as a model for creating these requirements. The purpose of this family 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 ADP_SCN_EX.1 explicitly stated SFR has dependencies on FPT_STM.1. 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 family since it is directly related to the main scanning functions of the TOE.

Given that ADP_RDR_EX.1 and ADP_SCN_EX.1 are similar to existing CC requirements and represent functions that would be evident in an applicable TOE, the assurance requirements of the CC should be appropriately applicable without any changes or additional guidance for their application. ADP_PRT_EX.1 is straight forward functions whose operation can be observed and as such is also evaluatable using the assurance requirements defined in the CC.

8.7 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.

	DB Scan (EXP)	**Security audit**	**Security management**
ADP_RDR_EX.1	X		
ADP_SCN_EX.1	X		
ADP_PRT_EX.1	X		
FAU_GEN.1a		X	
FMT_SMF.1			X

Table 6 Security Functions vs. Requirements Mapping

8.8 PP Claims Rationale

See Section 7, Protection Profile Claims.