National Information Assurance Partnership

TM

Common Criteria Evaluation and Validation Scheme Validation Report

Bit9, Inc. Parity™ 6.0.1

Report Number: CCEVS-VR-VID10436-2011
Dated: February 23, 2011
Version: 2.0

National Institute of Standards and Technology
Information Technology Laboratory
100 Bureau Drive
Gaithersburg, MD 20899

National Security Agency
Information Assurance Directorate
9800 Savage Road STE 6940
Fort George G. Meade, MD 20755-6940

Table of Contents

1 Executive Summary

The Target of Evaluation (TOE) is Bit9, Inc. Parity™ 6.0.1. The TOE was evaluated by the Booz Allen Hamilton Common Criteria Test Laboratory (CCTL) in the United States and was completed in February 2011. The evaluation was conducted in accordance with the requirements of the Common Criteria, Version 3.1 Revision 3 and the Common Methodology for IT Security Evaluation (CEM), Version 3.1 Revision 3. The evaluation was for Evaluation Assurance Level 2 (EAL2) augmented with ALC_FLR.1 (Flaw reporting procedures) and ASE_TSS.2 (TOE summary specification with architectural design summary). The evaluation was consistent with National Information Assurance Partnership (NIAP) Common Criteria Evaluation and Validation Scheme (CCEVS) policies and practices as described on their web site (http://www.niap-ccevs.org/).

Bit9, Inc. Parity™ Version 6.0.1 is a policy-driven whitelisting solution for restricting the execution of applications and devices that runs on modern Windows operating systems. Whitelisting technology allows end-users to install and run legitimate software and devices while providing information technology (IT) groups with a way to prohibit anything unauthorized or known to be malicious from executing. The end result is granular control of Windows computers, dramatically improving security, preventing software drift, and managing the flow of information to portable storage devices.

Parity's management capabilities track portable executable (PE) and script files and monitor their prevalence and execution. Unidentified files that have just appeared on the network receive a pending status. A file keeps its pending status until it becomes approved or banned. A pending file also can be acknowledged, which removes it from the list of new pending files but does not change its underlying pending status. Once a file is approved, it is allowed to execute on all systems but continues to be tracked.

After a network is under Parity control, Administrators approve new applications or patches using the approval methods that best suit their organization's software rollout procedures. Parity features several automatic approval methods (trusted directory, trusted publisher, trusted user, and trusted updaters) that make it easy to approve new software without having to do it file-by-file.

The Bit9 Parity software, when configured as specified in the installation guides and user guides (see Section 8 for necessary guidance), satisfies all of the security functional and assurance requirements stated in the TOE's Security Target.

The cryptography used in this product and its intended operational environment has not been FIPS-certified, nor has it been analyzed or tested to conform to cryptographic standards during this evaluation. All TOE cryptography has only been asserted as tested by the vendor. Note that the TOE does not provide the cryptography implementation used to protect data during transmission which is provided by the operational environment.

The technical information included in this report was largely derived from the Evaluation Technical Report (ETR) and associated test reports produced by the evaluation team. The Bit9, Inc. Parity™ 6.0.1 Security Target version 2.0, dated 22 February 2011 identifies

the specific version and build of the evaluated TOE. This Validation Report applies only to that ST and is not an endorsement of the Parity appliance by any agency of the US Government and no warranty of the product is either expressed or implied.

2 Evaluation Details

Evaluated Product	Bit9, Inc. Parity™ 6.0.1, when configured per the instructions in the 'Evaluated Configuration for Bit9 Parity 6.0.1' document
Sponsor & Developer	Bit9, Inc., Waltham, MA
CCTL	Booz Allen Hamilton, Linthicum, Maryland
Completion Date	February 2011
CC	*Common Criteria for Information Technology Security Evaluation*, Version 3.1 Revision 3, July 2009
Interpretations	None.
CEM	*Common Methodology for Information Technology Security Evaluation*, Version 3.1 Revision 3, July 2009
Evaluation Class	EAL2 Augmented ALC_FLR.1 and ASE_TSS.2
Description	The TOE is the Parity software, which is a security software product developed by Bit9, Inc. as a system access control product.
Disclaimer	The information contained in this Validation Report is not an endorsement of the Parity product by any agency of the U.S. Government, and no warranty of the system access control product is either expressed or implied.
PP	None
Evaluation Personnel	Emmanuel Apau Christopher Gugel Johnpaul Martin Jeremy Sestok Derek Scheer John Schroeder Amit Sharma
Validation Body	NIAP CCEVS Daniel P. Faigin, The Aerospace Corporation Jim Brosey, Orion Security

3 Identification

The product being evaluated is Bit9, Inc. Parity™ 6.0.1, when configured per the instructions in the '*Evaluated Configuration for Bit9 Parity 6.0.1*' document.

4 Security Policy

4.1 User Data Protection

The primary purpose of the TOE is to enforce access control policies against distributed Windows computers. Subjects who access these computers are known as Client Users. These access control policies are centrally defined on the Parity Server and distributed to systems in an enterprise. The policies are subsequently enforced by an instance of the Parity Client which resides on each system. The effect of applying these access control

policies is known as "enterprise whitelisting". Whitelisting marks various specific files, processes, registry values, and removable media devices as authorized to be modified. The specific notion of enterprise whitelisting refers to the fact that individual users and computers in an enterprise can have different policies enforced upon them based on organizationally maintained Active Directory information.

Policies also contain a value called a SecCon, which is short for "security condition". The SecCon value of a policy determines what actions to take when an operation is performed on a system which is not whitelisted. For example, under the most restrictive setting, actions that are not whitelisted are forbidden outright. Another setting warns the subject that they are performing an action that has not been approved and gives them an option to proceed or abort using a dialog box.

The Parity Server, which is where policies are defined and distributed, maintains its own role-based access control policy. Subjects which can access the Parity Server are called Console Users and can be assigned one of three specific roles: Administrator, PowerUser, and ReadOnly. Each of these roles confers a fixed set of privileges on the subject. This ensures that only trusted subjects are able to configure the TOE's behavior.

4.2 Security Audit

The TOE collects, aggregates, and reports on IT activity and generates alerts when file/device information changes. "IT activity" refers the content and behavior of systems that reside in the operational environment. Auditing functions are performed by the TOE by collecting the logs via agents and servers, using a database in the operational environment to store the logs over an established timeframe, and presenting the logs via reports and queries. In addition, the TOE generates audit reports for its own startup and shutdown and all user actions on the TOE. Authorized users are able to select the notification mechanism for all auditable events. The TOE monitors these for events and notifies (alerts) users when a predefined condition is met. Alerts can be sent via email, which requires mediation by an environmental Simple Mail Transfer Protocol (SMTP) server. The TOE relies on the host operating system to provide reliable timestamps for audit records.

4.3 Identification and Authentication

The TOE supports two types of users: Console Users and Client Users.

Console Users manage the TOE remotely through a Web Browser. They are identified and authenticated with username and password. This authentication data can be maintained within the TSF, or Active Directory integration can be used to allow an existing organizational user to access the TOE using credentials maintained by the operational environment. If the operational environment is used, username/password validation will be done with Lightweight Directory Access Protocol (LDAP).

Client Users access the TOE indirectly by using their own local machines upon which the Parity Client has been installed. Modifications to their machines are mediated by the TOE, but the TOE is invisible except for pop-up messages when an operation has been blocked. In the evaluated configuration, Client Users are identified by the user account they use to log in to their system, which is derived from Active Directory.

4.4 Security Management

There are four default roles for the TOE: Administrators, PowerUsers, ReadOnly, and Client Users. Administrators, PowerUsers, and ReadOnly are all types of Console Users. The role given to a user determines what operations or management functions they can perform on the TOE. Restrictions can be set on Client Users based on policy; permissions for Console Users are static.

The major security management functions of the TOE are the ability to review audit data and the ability to configure how the client access control policy is enforced. Policy data is propagated to clients and stored internally. The Parity Console, as a web-based application, requires the use of an environmental Domain Name Service (DNS) server if it is to be identified by a qualified domain or host name as opposed to an Internet Protocol (IP) address.

4.5 TOE Access

Before a session begins, a warning will be displayed alerting the Console User that unauthorized access to the TOE is prohibited.

4.6 Encrypted Communications

Remote users establish a session with the Parity Server using a web-based GUI that is secured via Hypertext Transfer Protocol Secure (HTTPS). Cryptography for this is provided by the environmental web server and operating system. This secured path is used for user authentication and management of the TOE by authorized users. The operational environment generates cryptographic keys during communication with remote users, the Open Database Connectivity (ODBC) client, and Active Directory. This is accomplished by the TOE requesting that these environmental components use their native cryptographic facilities. All communications between the Parity Server and Parity Clients and between the Parity Server and the Parity Global Software Registry (GSR) are protected using imported certificates.

The only cryptographic function provided by the TSF is the ability to hash files for the purpose of access control checking.

The cryptography used in this product has not been FIPS-certified, nor has it been analyzed or tested to conform to cryptographic standards during this evaluation. All cryptography has only been asserted as tested by the vendor.

5 Threats/Assumptions/Organizational Security Policies

5.1 Threats to Security

The product addresses the following threats:

- Unauthorized users could gain local or remote access to protected objects that they are not authorized to access.
- An administrator may incorrectly install or configure the TOE or install a corrupted TOE, resulting in ineffective security mechanisms.

- A malicious user or process may view audit records, cause audit records to be lost or modified, or prevent future audit records from being recorded, thus masking a user's action.
- A malicious user or process may impersonate the Global Software Registry (GSR) or Parity Application Server in order to intentionally provide inaccurate configuration information or metadata to the TOE.
- A malicious or ignorant user may acquire and configure a reverse-engineered version of the TOE that bypasses or subverts access control to protected resources.
- Malicious or non-malicious users could gain unauthorized access to the console by bypassing identification and authentication countermeasures.

5.2 Connectivity Assumptions

The product makes the following connectivity assumptions:

- The TOE will be deployed in an environment where external data stores reside on a trusted network and client systems have the capability to communicate with the Parity Application Server intermittently if not persistently.
- Clients deployed on remote systems will be installed in a context that prevents Client Users from disabling, removing, altering, or reconfiguring the client.
- Client Users are identified to the TOE via the host name of their workstation and/or the Active Directory credentials used to authenticate to it.

5.3 Personnel Assumptions

The product makes the following assumptions regarding personnel:

- One or more authorized administrators will be assigned to install, configure and manage the TOE.
- Administrators and PowerUsers of the TOE are not careless, willfully negligent, or hostile and will abide by the instructions provided by applicable guidance documentation.
- Administrators exercise due diligence to update the Operational Environment with the latest patches in order to remove the risk of compromise via known and preventable exploits.
- Console Users will either choose strong passwords as defined by their organizational guidance or, if Active Directory integration is used for the Parity Console, that the Active Directory enforces strong password policies.

5.4 Physical Assumptions

The product makes the following assumptions regarding the physical environment:

- The TOE server and remote database will be located within controlled access facilities that will prevent unauthorized physical access.

5.5 Organizational Security Policies

The following organizational security policies are assumed to be applicable:

- The TOE shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which users consent by accessing the system.

6 Clarification of Scope

All evaluations (and all products) have limitations, as well as potential misconceptions that need clarifying. This text covers some of the more important limitations and clarifications of this evaluation. Note that:

1. As with any evaluation, this evaluation only shows that the evaluated configuration meets the security claims made, with a certain level of assurance (EAL 2 in this case).
2. This evaluation only covers the specific version of the product identified in this document, and not any earlier or later versions released or in process.
3. As with all EAL 2 evaluations, this evaluation did not specifically search for, nor seriously attempt to counter, vulnerabilities that were not ―obvious‖ or vulnerabilities to objectives not claimed in the ST. The CEM defines an "obvious" vulnerability as one that is easily exploited with a minimum of understanding of the TOE, technical sophistication and resources.

6.1 TOE

The evaluated configuration of the TOE includes the Bit9, Inc. Parity™ 6.0.1 product, which consists of the following components:

- Parity Application Server
- Parity Client

The following features of the TOE have been excluded from the evaluation; review ST Section 2.4.9 for more information:

- Use of the command line interface (CLI)
- Windows Embedded for Point of Service (WEPOS)
- Live Inventory Software Development Kit (SDK)

6.2 Operational Environment

6.2.1 Software Requirements

The following operational environment software is considered to be outside the evaluation boundary:

- Microsoft Internet Information Server (IIS) 6.0 or higher
- Microsoft Internet Explorer 7.0 or higher
- Microsoft Windows XP SP2 or higher (for clients)
- Microsoft Windows Server 2003 or 2008 SP2 (for the Application Server)

- Mozilla Firefox 3.0 or higher
- SMTP Server – any server that allows unauthenticated SMTP connections
- Active Directory Windows 2003 SP2 or higher
- Syslog server – any server that allows unauthenticated Syslog connections
- Microsoft SQL Server (regular or Express) version 2005 r2 or higher
- VMware ESX Server 4.0
- LDAP server (for Active Directory) – version that is compatible with Active Directory Windows 20003 SP2 or higher
- DNS server – any DNS server is acceptable
- Parity Knowledge (unversioned, the TOE will automatically connect to the most up-to-date server)

6.2.2 System Requirements

The following components are recommended on the appliances for the TOE:

Parity Application Server

Parity Client Load	Less than 300 Clients	300 to 50000 Clients
Processor	Dual Core Server Class	Dual Core or Dual Processor Server Class
Disk space	40 GB	2 drives: 40GB for Parity, 72GB for SQL
RAM	2-4 GB	4 GB
Network	1 GB NIC	1 GB NIC
IP address	Fixed IP address or (preferably) a fully qualified DNS name. Computers running the Parity Client recognize the server by either its fixed IP address or DNS-name lookup.	
Operating System	• Microsoft Windows Server 2003 - SP2 32-bit, with Microsoft Internet Information Services (IIS) version 6.0, with latest patches. • Microsoft Windows Server 2008 SP2 32-bit or 64-bit, with Microsoft Internet Information Services (IIS) version 7.0, with any patches. • For both Server 2003 and 2008, install .NET 3.5, with latest patches.	
SQL Server	• SQL Server 2005 Express SP2 for <300 Client computers • SQL Server 2005 SP2 or above Standard/Enterprise OR SQL Server 2008 SP1 or above for >300 Client Computers	

Parity Application Server Virtualization

To run VMware ESX Server v.4.0+ to create a virtualized environment for Parity:

- Memory meeting the configurations must be allocated as ‘reserved’
- Minimum dual virtual processors are required for all configurations

The hardware requirements for the machine hosting the virtual environment should allow the virtual environment to be capable of the same level of performance as a hardware-based installation.

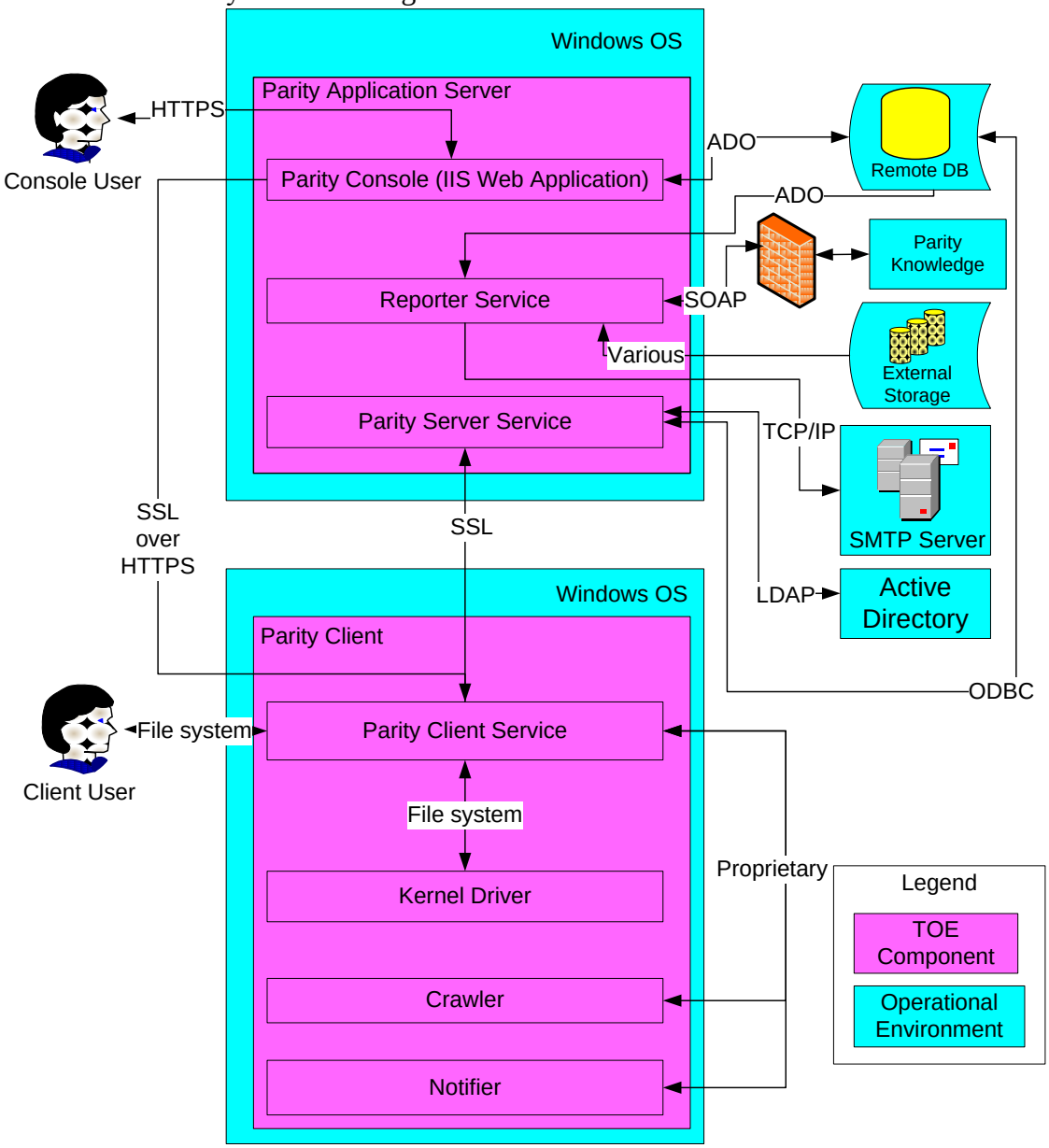
Parity Client

OS (all versions of:)	Processor	CPU	RAM	Disk Space

Windows XP 32-bit, SP2 & SP3	Intel Pentium 4 650MHz (or equivalent processor)	650 MHz	Any configuration that enables standard desktop applications to run with good performance, preferably at least 768MB.	Approximately 65 MB, depending on the number of applications installed on the system
Windows 2003 Server 32-bit & 64-bit & R2, SP1				
Windows Vista 32-bit & 64-bit, SP1 & SP2				
Windows 2008 Server 32-bit & 64-bit, Windows 2008 Server R2 64-bit				
Windows 7 32-bit & 64-bit				

7 Architectural Information

The TOE's boundary defined in Figure 1.



Acronyms:
ADO - ActiveX Data Objects
SOAP - Simple Object Access Protocol
SMTP - Simple Mail Transfer Protocol
LDAP - Lightweight Directory Access Protocol
HTTPS - Hypertext Transfer Protocol Secure
SSL - Secure Sockets Layer
TCP/IP - Transmission Control Protocol/Internet Protocol

Figure 1 – TOE Boundary for Bit9, Inc. Parity™ 6.0.1

7.1 TOE Components

7.1.1 Parity Application Server

Parity Application Server software runs on standard Windows computers. It can be run on a dedicated system or on a virtual machine. Parity Application Server is used to manage policies, including software and device approvals and bans, and to provide visibility into events and file activity on computers running Parity Clients.

Using HTTPS, Console Users access the Parity Application Server and the TOE through a web browser. Console User access to the Parity Application Server is determined by the Parity Console (IIS Web App). The Parity Console (IIS Web App) performs the identification and authentication of all users of the Parity Application Server. All administration and reporting functions, to include access information, of Parity are communicated directly with the environmental database using ADO.

Console User authentication uses Parity Server Service and a secure LDAP connection with Active Directory to validate credentials and look up account memberships. Using ODBC, Parity Server Service logs record session information or errors to the database.

Once a user has successfully authenticated to TOE via the Parity Console, the Console User must then be authorized to perform actions on the TOE by the Parity Server Service. The most important action which can be authorized by the Parity Server Service is the ability to create policies. Examples of policies include but are not restricted to: (1) approving a discovered file by its hash value, (2) approving a discovered file by trusted publisher (digital certificate), (3) blocking registry changes, (4) banning a discovered file by its hash value, (5) approving a device for network use, or (6) trusting all content of a client directory.

After policy creation and when a connection between Parity Client and Parity Server is re-established after being disconnected, the Parity Server Service will propagate polices to Parity Clients using operational environment provided SSL. In addition to passing client policies to Parity Clients, the Parity Server Service also receives events using operational environment provided SSL.

Note that the product's SSL implementation is provided by the operational environment.

Additionally, the Reporter Service handles all scheduled and background tasks for the Parity system. It uses a mutually authenticated web service (Simple Object Access Protocol (SOAP) over operational environment provided SSL) connection with Parity Knowledge/Global Software Registry (GSR) to retrieve enhanced metadata regarding all files known to the system. This enhanced metadata includes threat, trust, and categorization values. The Reporter Service accesses the Parity database directly using ADO.

Note: Although only Console Users authenticated by the TOE may view audited events through database queries, alerts may be sent to any valid email address.

Lastly, Parity tracks executable files and monitors their prevalence and execution. Unidentified files that have just appeared on the network receive a pending status. A file keeps its pending status until it becomes approved or banned. A pending file also can be

acknowledged, which removes it from the list of new pending files but does not change its underlying pending status. Once a file is approved, it is allowed to execute on all systems but continues to be tracked.

Besides blocking unauthorized files, Console Users can use other Parity features to determine information such as the following:

- Whether a file exists on a computer on the network
- Which computers have the file
- Where and when the file first arrived in the network environment
- What is known about the source, category, trust level, and threat of the file
- Whether and when a file has executed, and on which computers
- How the inventory of files on computers has changed over time
- Whether a file has propagated and, if so, whether it has been renamed
- Whether certain USB storage devices exist on a network, when they first were discovered, and on what computer

7.1.2 Parity Client

Parity Client, also known as Parity Service Agent, software running on client computers monitors files, process and registry activity and communicates with the Parity Application Server when necessary. If the client is unable to connect to the Parity Server, it uses an offline policy to make decisions that allow for uninterrupted enforcement of access control. When a disconnected computer running the Parity Client reconnects, the agent receives updates from the server and communicates relevant file activity during the time it was disconnected from the network. The Parity Client runs silently in the background until it blocks a file, at which point it displays a message to the computer user that explains why the file was not permitted to execute. Depending on the file state and the agent's security level, Parity may be configured to let the user on the client computer choose to run a blocked file.

8 Documentation and Delivery

The NIAP-certified Bit9 Parity product is acquired via normal sales channels, and delivery of the TOE is coordinated with the end customer by Bit9, Inc. Parity 6.0.1 (to include Parity Application Server and Parity Client). The product is provided to normal customers through physical and electronic delivery as a software package accompanied by the following set of documents.

- "*Using Parity v6.0.1*", November 29, 2010, document version 6.0.1.b.
- "*Installing Parity v6.0.1*", November 29, 2010, document version 6.0.1b.
- "*Operating Environment Guidelines: Parity™ Version 6.0.1*", October 15, 2010.
- "*Introduction to Bit9 Parity v6.0*", July 9, 2010 version 1.0
- "*Parity 6.0.1 Release Notes*", November 30, 2010 version 1.2

Additionally, Bit9 provides documentation on their support website. Included within this documentation is the '*Evaluated Configuration for Bit9 Parity 6.0.1*' which must be referenced to place the product within the CC evaluated configuration.

The following documents were used as evaluation evidence. Proprietary documents (i.e., those not available to the general public) are indicated with ‡.

- Guidance Documentation (AGD)
  - "*Using Parity v6.0.1*", November 29, 2010
  - "*Installing Parity v6.0.1*", November 29, 2010
  - "*Operating Environment Guidelines: Parity™ Version 6.0.1"*, October 15, 2010
  - "*Evaluated Configuration for Bit9 Parity 6.0.1",* no date
- Security Target (ST)
  - "*Bit9 Parity v6.0.1 Security Target*", February 22, 2011 v2.0
- Development (ADV) Evidence Documentation
  - "*TOE Design Specification for Bit9 Parity 6.0.1*", February 22, 2011 v2.0‡
  - "*Functional Specification for Bit9 Parity 6.0.1*", February 22, 2011 v2.0‡
- Life-Cycle (ALC) Evidence Documentation
  - "*Bit9 - Source Code Management System*", November 18, 2010 v1.1‡
  - "*svn-log-Trillian.txt"* ‡
  - "*svn-log-Trillian-M1.txt"* ‡
  - *"Trillian-6.0-Files.txt"* ‡
  - "*Trillian-6.0-M1-Files.txt"* ‡
  - "*Bit9 - Delivery Evidence*", November 24, 2010 v1.4 ‡
  - "*Bit9 - Incident (Flaw) Remediation*", December 20, 2010 v1.3‡
- Testing (ATE) and Vulnerability Analysis (AVA) Documentation
  - "*Common Criteria Test Plan – 6.0.1"*, 11/7/2010 ‡
  - "*Common Criteria Test Plan*" (matrix) ‡
  - "Vendor_Results" (matrix)
  - "Booz Allen_Bit9_Parity_INDTestProcedures" (matrix)

    ‡

- "*Bit9, Inc. Parity™ 6.0.1Evaluation Team Test Report*", Version 3.0, October 6, 2010‡
- "*Vulnerability Analysis Bit9, Inc.Parity™ 6.0.1*", Version 3.0, February 22, 2011‡

9 IT Product Testing

9.1 Functional Testing

9.1.1 Functional Test Methodology

The evaluation team's test approach was to test the security mechanisms of the Bit9 Parity 6.0 by exercising the external interfaces to the TOE and viewing the TOE behavior on the platform. Each TOE external interface was described in the relevant design documentation (e.g., FSP) in terms of the relevant claims on the TOE that can be tested through the external interface. The ST, TOE Design (TDS), Functional Specification (FSP), Security Architecture (ARC) and the vendor's test plans were used to demonstrate test coverage of all EAL2 requirements for all security relevant TOE external interfaces. TOE external interfaces that will be determined to be security relevant are interfaces that perform any of the following:

- Change the security state of the product
- Permit an object access or information flow that is regulated by the security policy
- Are restricted to subjects with privilege or behave differently when executed by subjects with privilege
- Invoke or configure a security mechanism

Security functional requirements were determined to be appropriate to a particular interface if the behavior of the TOE that supported the requirement could be invoked or observed through that interface.

9.1.2 Functional Results

During the course of the evaluation, the Booz Allen evaluation team reviewed the vendor's functional testing and determined that all security relevant TOE external interfaces were tested and a majority of the claimed functionality was tested by the vendor. The evaluation team then created a test plan that contained a sample of the vendor functional test suite, and supplemental functional testing developed by the evaluators. The evaluators test suite emphasized on the product's primary functionality, any claimed functionality not fully covered by the vendor's test suite, and additional regression testing. Based upon the results of the vendor and evaluator testing; it has been determined that the product functionally operates as described.

9.2 Vulnerability Testing

9.2.1 Vulnerability Test Methodology

The evaluation team created a set of vulnerability tests to attempt to subvert the security of the TOE. These tests were created based upon the evaluation team's review of the vulnerability analysis evidence and independent research. The Evaluation Team conducted searches for public vulnerabilities related to the TOE. A few notable resources consulted include securityfocus.com, the cve.mitre.org, and the nvd.nist.gov.

Upon the completion of the vulnerability analysis research, the team had identified several generic vulnerabilities upon which to build a test suite. These tests were created specifically with the intent of exploiting these vulnerabilities within the TOE or its configuration.

The team tested the following areas:

- **Eavesdropping on Communications**

  In this test, the evaluators manually inspected network traffic to and from the TOE in order to ensure that no useful or confidential information could be obtained by a malicious user on the network.

- **Port Scanning**

  Remote access to the TOE should be limited to the standard TOE interfaces and procedures. This test attempted to find ways to bypass these standard interfaces of the TOE and open any other vectors of attack.

- **Vulnerability Scanner (Nessus)**

  This test used the Nessus Vulnerability scanner to test any and all open interfaces on any applicable systems of the TOE. The scanner probes a wide range of vulnerabilities that includes but is not limited to the following:

Backdoors	Gain root remotely	RPC
CGI abuses	General	Settings
Denial of Service	Miscellaneous	SMTP Problems
Finger abuses	Netware	SNMP
Firewalls	NIS	Untested
FTP	Port scanners	Useless services
Gain a shell remotely	Remote file access	

- **Unauthenticated Access / Directory Traversal Attack**

  This test used “URL hacking” to attempt to access protected TOE resources by injecting unexpected input into requests that were sent to the TOE. This was done using two different approaches to URL exploitation.

  - The first part attempted to access protected TOE resources as an unauthenticated outsider.

- The second part attempted to access local TOE resources that should be protected from any remote access (unauthenticated and authenticated).

- **SQL Injection / Cross Site Scripting Attack / Cross Site Request Forgery**

  This test executed automated SQL Injection and Cross Site Scripting attacks against the TOE. The evaluators determined any fields or variables that could be prone to attack. They then used a scanner, which contained a large database of standard strings that are used for testing SQL Injection and Cross Site Scripting issues. These strings were input into the various fields and variables and the output was analyzed for inconsistencies.

- **Web Server Vulnerability Scanner (Nikto)**

  This test used the Nikto web server vulnerability scanner to test for any known vulnerabilities that could be present in the TOE's web interfaces. This scanner probed a wide range of vulnerabilities that included the following:

File Upload	Denial of Service
Interesting File / Seen in logs	Command Execution / Remote Shell
Misconfiguration / Default File	SQL Injection
Information Disclosure	Authentication Bypass
Injection (XSS/Script/HTML)	Software Identification
Remote File Retrieval	Remote source inclusion

- **Windows Startup Sequence**

  The Bit9 executable on the client machine was configured to run when the client machine performs its initial boot. This test attempted to remove the Bit9 executable from the startup sequence thereby forcing the system to start in an insecure manner.

- **Configuration File Tampering**

  This test attempted to modify resources that Bit9 relies upon for configuration and operational parameters. If an un-trusted user was allowed to modify these resources, it could have been possible to interrupt or subvert the security functionality of the TOE.

- **Alternate Data Streams**

  This test attempted to hide a non-permitted executable inside an alternate data stream behind a permitted executable. Executables in alternate data streams can be executed from the command line and they will not show up in Windows Explorer. If the non-permitted executable was able to be hidden from the Bit9 Client, or the Client could be tricked into thinking that the malicious executable was trusted, then ADS could have been a way to bypass the restrictions of the TOE.

- **DLL Injection**

  This test attempted to inject a non-trusted DLL into a trusted executable. After the DLL was injected, its potentially malicious code could have run before jumping to the main function of the trusted executable. If this code was allowed

to run in the context of the trusted executable, then this could have represented a way to bypass the restrictions of the TOE.

- **Embedded Code and Process Migration**

  This test attempted to run meterpreter as a malicious ActiveX script inside of Internet Explorer (trusted) and then attempted to use the meterpreter migration functionality to migrate to different trusted running process.

9.2.2 Vulnerability Results

During the vulnerability testing, there were several issues discovered that could affect the security posture of a deployed system. These issues have been broken up into the following categories:

9.2.2.1 Mitigated by Guidance

- **Weak SSL Ciphersuites Used**

  The TOE's environment uses weak SSL ciphersuites by default. IIS must be used to enable stronger SSL ciphersuites. Section 4.4 of '*Evaluated Configuration for Bit9 Parity 6.0.1*' details how to configure the TOE's environment to use stronger SSL ciphersuites.

- **Product Does Not Automatically Verify Certificates Between Agent and Server**

  The TOE does not automatically verify certificates between the Agent and the Server. This can lead to a man-in-the-middle attack between the Agent and Server. This option needs to be enabled after the installation of the TOE. Section 4.4 of '*Evaluated Configuration for Bit9 Parity 6.0.1*' details how to configure the TOE to verify these certificates.

- **Product Can Be Configured to Ban Files on Filename**

  The TOE can be configured to ban/approve files based on filename rather than file hashes. This is dangerously insecure as malicious files can masquerade as legitimate files within the filename, or legitimate files may be altered to contain malicious DLLs. Section 4.5 of '*Evaluated Configuration for Bit9 Parity 6.0.1*' advises against using this option .

10 Results of the Evaluation

The evaluation was carried out in accordance with the Common Criteria Evaluation and Validation Scheme (CCEVS) process and scheme. The evaluation demonstrated that the Bit9, Inc. Parity™ 6.0.1 TOE meets the security requirements contained in the Security Target.

The criteria against which the Parity TOE was judged are described in the Common Criteria for Information Technology Security Evaluation, Version 3.1 Revision 3, July 2009. The evaluation methodology used by the evaluation team to conduct the evaluation

is the Common Methodology for Information Technology Security Evaluation, Version 3.1 Revision 3, July 2009. The Booz Allen Hamilton Common Criteria Test Laboratory determined that the evaluation assurance level (EAL) for the Bit9, Inc. Parity™ 6.0.1 TOE is EAL2 augmented with ALC_FLR.1 and ASE_TSS.2. The TOE, configured as specified in the installation and configuration guides, satisfies all of the security functional requirements stated in the Security Target.

The evaluation was completed in February 2011. Results of the evaluation and associated validation can be found in the Common Criteria Evaluation and Validation Scheme Validation Report.

11 Validator Comments/Recommendations

11.1 Secure Installation and Configuration Documentation

The "*Evaluated Configuration for Bit9 Parity 6.0.1*" document defines the recommendations and secure usage directions for the TOE as derived from testing. It must be downloaded and consulted during installation.

11.2 STIG Compliance

During the course of the evaluation the TOE and its operational environment's ability to conform to applicable Defense Information Security Administration (DISA) Security Technical Implementation Guides (STIGs) was not verified.

11.3 SMTP Servers with Authentication

The TOE does not provide mechanisms to allow it to communicate with SMTP Servers that require authentication.

11.4 Password and Login Frustration Mechanisms

The TOE does not provide mechanisms that require strong passwords and login frustration. For this reason, the assumption A.PASSWORD was included in the evaluation. The assumption states, "Console Users will either choose strong passwords as defined by their organizational guidance or, if Active Directory integration is used for the Parity Console, that the Active Directory enforces strong password policies."

11.5 Reliance on Operational Environment's MSI Installation Mechanism

The TOE's constructed installers for the client machines are Microsoft Windows Installer (MSI) files. Since these MSI files may be installed after the machine is already in an operational state, and thus there is a dependence on the operational environment's MSI installation mechanism for correct enforcement of policy on the clients.

11.6 Verify Values Inherited from Existing Policies

Parity includes a built-in policy called the Template Policy. As the name suggests, this is a "template" for creating other policies. By default, the initial settings of the first policy a user creates are based on the settings of this Template Policy, although a user can also choose to use any other existing policy, including the Default Policy. When a user creates a new policy, they must verify the values or, if needed, change the setting values inherited from the existing policy upon which they based it.

11.7 SMTP and Syslog Protocols

Bit9 Parity contains functionality to send alerts remotely via email (SMTP), and to send audit data remotely via syslog. These protocols are used for simplicity and efficiency. However, it should be noted that these protocols transmit information over the network in an unencrypted format, which could be susceptible to a compromise of confidentiality or integrity if appropriate precautions are not taken.

If the remote email or logging functions are used, the TOE should be deployed only in trusted network environments where there is reasonable assurance about the identity of all connected machines and corresponding users. In addition, there should be reasonable assurance that all connected machines and corresponding users are non-malicious in nature.

11.8 Applicability to NIST SP 800-53/CNSS 1253

The use of Parity partially satisfies the NIST SP 800-53 revision 3 (as completed by CNSSI 1253) security control of AU-2. Parity defines a number of auditable events for activities performed both against itself and against operational environment systems to which access control policies are applied. These auditable events allow for administrative visibility into the types of activities that are performed within the enterprise. This is considered to be a partial satisfaction of AU-2 because the TSF is only responsible for auditing actions against itself and against file, program, registry, and removable device activity in the operational environment. It is not responsible for auditing other activities such as Client User authentication or credential management, for example.

12 Security Target

The security target for this product's evaluation is *"Bit9, Inc. Parity™ 6.0.1 Security Target"*, Version 2.0, dated 22 February 2011.

13 List of Acronyms

Acronym	Definition
AD	Active Directory
ADO	ActiveX Data Objects
ADS	Alternate Data Streams
ADSI	Active Directory Service Interfaces
API	Application Programming Interface
ARC	Security Architecture
CA	Certificate Authority
CC	Common Criteria
CCEVS	Common Criteria Evaluation and Validation Scheme
CCTL	Common Criteria Test Laboratory
CEM	Common Methodology for IT Security Evaluation
CLI	Command Line Interface
CNSSI	Committee on National Security Systems Instruction
COM	Component Object Model
DISA	Defense Information Security Administration
DLL	Dynamic-link library
DNS	Domain Name Service
EAL	Evaluation Assurance Level

ETR	Evaluation Technical Report
FIPS	Federal Information Processing Standard
FSP	Functional Specification
FTP	File Transfer Protocol
GB	Gigabyte
GSR	Global Software Registry
HTTPS	Hypertext Transfer Protocol Secure
IIS	Internet Information Services
IP	Internet Protocol
IT	Information Technology
LDAP	Lightweight Directory Access Protocol
MB	Megabyte
MDAC	Microsoft Data Access Component
MSI	Microsoft Windows Installer
NIAP	National Information Assurance Partnership
NIC	Network interface card
NIS	Network Information Service
NIST	National Institute of Standards and Technology
ODBC	Open Database Connectivity
OLE	Object Linking and Embedding
OS	Operating System
PE	Portable Executable
PP	Protection Profile
RPC	Remote procedure call
SAR	Security Assurance Requirements
SDK	Software Development Kit
SecCon	Security Condition
SFR	Security Functional Requirements
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
SOAP	Simple Object Access Protocol
SQL	Structured Query Language
SP2	Service pack 2
SSL	Secure Sockets Layer
ST	Security Target
STIG	Security Technical Implementation Guide
TCP/IP	Transmission Control Protocol/Internet Protocol
TDS	TOE Design
TLS	Transport Layer Security
TOE	Target of Evaluation
TSF	TOE Security Functions
TSS	TOE Summary Specification
URL	Uniform Resource Locator
USB	Universal Serial Bus
UTC	Universal Time Code
WEPOS	Windows Embedded for Point of Service
XP	Windows operating system
XSS	Cross site scripting

14 Terminology

Terminology	Definition
.NET	A software framework that can be installed on computers running Microsoft

	Windows operating systems. It includes a large library of coded solutions to common programming problems and a virtual machine that manages the execution of programs written specifically for the framework. The .NET Framework is a Microsoft offering and is intended to be used by most new applications created for the Windows platform.
Active Directory (AD)	Technology created by Microsoft that provides a variety of network services, including: • Lightweight Directory Access Protocol (LDAP)-like directory services • Kerberos-based authentication • DNS-based naming and other network information • Central location for network administration and delegation of authority • Information security and single sign-on for user access to networked based resources • Central storage location for application data • Synchronization of directory updates amongst several servers
ActiveX Data Objects (ADO)	A set of Component Object Model (COM) objects for accessing data sources. A part of Microsoft Data Access Component (MDAC), it provides a layer between programming languages and Object Linking and Embedding (OLE) Database (a means of accessing data stores, whether they be databases or otherwise, in a uniform manner).
Administrator	Monitors file activity and configures all aspects of the system, including creating policies, setting alerts, and creating all types of user accounts
Alert	A tool that provides notifications in the Parity Console and email notifications to a list of subscribers when a specific event an Administrator or PowerUser may choose occurs. Alerts can be made policy-specific.
Baseline	A reference point that can be used to determine drift of computers running Parity Client from the reference, and thus potential risk for those computers. A baseline can be a named table of files or the current set of files on a reference computer.
Basic Authentication	Authentication to the TOE through the use of any valid user interface. Data for credentials are stored on the database rather than Active Directory.
Blacklist	A list or collection of software that, for one reason or another, is expressly forbidden.
Client Access Policies	The union of Parity Policies (see "Policy), Policy Settings (see "Policy Settings"), and Policy Rules. Client Access policies include the information contained within the Policy Setting definition below as well as the more granular approval/ban rules covering individual devices, files, publishers, processes, applications, and users.
Client Users	Users who interact with the TOE via the Parity Client. Their requests to interact with their local system are mediated by the Client Access Policy.
Computer	Windows-compatible computer that runs the Parity Client. Each computer protected by Parity communicates with the Parity Application Server via the agent when it is on the same network.
Computer initialization	File initialization process for new computers that come online to the Parity system. During initialization, each file on the hard-drive of the new machine

	is evaluated and classified by the Parity Application Server.
Console Access Policies	The union of Parity Policies (see "Policy), Policy Settings (see "Policy Settings"), and Policy Rules. Console Access policies include information regarding the user's identification along with group association to determine what information can be accessed.
Console Users	Term that refers to the collection: Administrators, PowerUser, and ReadOnly. Collectively, these are the users which are permitted to access the management interface of the TOE.
Dashboard	Interface that graphically displays network environment information through a series of compact "portlets." Through a Dashboard, Parity Server users can drill down to more detailed information about network elements such as files, computers, and alerts.
Device Rules: Approvals	Console Users can choose to approve file activity on specific, detected USB devices, regardless of the security policy in force for a computer.
Drift	Information that can help determine to what extent one or more computers have changed from a baseline of files. This can help determine level of compliance with company policies on acceptable files, and also identify files that should be approved and added to an updated baseline.
Event	An event is informational message resulting from Parity activities that can be filtered and presented as custom reports. Events include files blocked, pending files executed, and changes made to the system by console users.
Executable	An executable is any file that contains executable code. Parity examines the *content* of each unknown file that appears on a computer in its network, determines whether it contains executable code, and if so, categorizes it according to executable type. Scripts are included in this process.
Explicit Group Assignment	Assignment to restrict/allow performable functions of the TOE. Falls under the different roles types as defined by the TOE.
File Rule	The Files tab of the Software Rules page shows all of the approvals and bans created at a site for individual files. From this tab, a Console User can manage rules. The Console User can check to see whether a particular file has any rule, file or ban, affecting it, and the Console User can remove rules from one or more checked files.
File state	Parity classification that determines how executables are tracked and permitted or not permitted to be run. File state includes approved, banned, and pending states.
Files	Display of the Files page, which includes tabbed lists of networked files, including: File Catalog, a searchable archive of all unique files hashed on the Parity server, and Files on Computers, a list of all files running on all of computers running Parity Agent.
Group	A set of users with a selection of permissions that is stored either in the Active Directory or a Parity-defined group.
Hypertext Transfer Protocol Secure (HTTPS)	A combination of the Hypertext Transfer Protocol with the SSL/TLS protocol to provide encryption and secure (website security testing) identification of the server.
Internet Information Services (IIS)	Set of Internet-based services for servers using Microsoft Windows
Lightweight Directory Access Protocol (LDAP)	An application protocol for querying and modifying data using directory services running over TCP/IP. A directory is a set of objects with attributes organized in a logical and hierarchical manner. A simple example is the telephone directory, which consists of a list of names (of either persons or organizations) organized alphabetically, with each name having an address and phone number associated with it.
Manifest file	A document that lists the contents and attributable file information that uniquely identifies each file contained in the document. This information may be used to automatically generate a set of approval policies.

Meter	A software tool that enable a Console User to monitor the number of executions of files an Administrator or PowerUser may specify, and the users and computers executing them.
Modes	Active Parity Agents can be operated in one of two modes: Visibility Only provides the file and event tracking features of Parity, but does not enforce file or device bans or other security restrictions. Visibility and Control mode blocks banned files and allows a Console User to choose one of three SecCons to determine how pending files are treated. Visibility and Control policies can be configured to enforce other file and device security rules.
Open Database Connectivity (ODBC)	Standard software API method for using database management systems (DBMS). The designers of ODBC aimed to make it independent of programming languages, database systems, and operating systems.
Parity Application Server	Windows-compatible computer running the Bit9 Parity Application Server software.
Parity Client	Agent software installed on computers on a network; the agent runs independently but reports to the Parity Application Server.
Parity Console	The console, which can be displayed remotely with a web browser, is the user interface and management center for all Parity management activities.
Policy	Each computer protected by Parity is associated with a policy that defines its security characteristics. Computers with the same security requirements can share the same policy. Note that users of computers running the Parity Client do not need Parity accounts. The server requires no direct interaction with users of computers Parity is monitoring.
Policy Settings	Policy settings define the way a Parity user manages a particular group of computers. There are three categories of settings: basic policy definitions, device settings, and advanced settings. • Basic policy definitions include the policy name and other descriptive information, whether computers in this policy allow agent upgrades, whether live file inventory is activated for these computers, and the basic security level (the Mode and SecCon) for the policy. • Device settings control the way a Parity policy treats removable devices. Parity Administrators or PowerUsers can make different rules to control read, write, and execute operations on devices, and can designate approved devices to be treated differently than non-approved devices. • Advanced policy settings control whether computers in a policy have certain file types blocked, whether files installed by specially designated "trusted" users are allowed to execute, and whether special treatment of certain directories is enabled. The possible values are Active, Off, and Report Only.
Portlet	A Dashboard element that displays information such as the number and types of computers managed by Parity, the number and type of security policies enforced, the drift of files on one or more computers away from a reference point, the types of software on a network, and whether the files identified by Parity are compatible with Vista. Although there is variation in the specific information displayed, their structure is generally similar. Each portlet has a banner with its name in the top left and a series of buttons in the top right.
PowerUser	A type of Console User which monitors file activity and can set policy. PowerUsers have limited user-account creation privileges, including creating ReadOnly and Unauthorized user accounts but not Administrator and other PowerUser accounts. Some system settings cannot be changed by PowerUsers.

ReadOnly	A type of Console User which monitors file activity and views previously executed Find Files search results. ReadOnly users cannot change any aspect of the Parity system configuration, and cannot create or edit users, policies, bans or approvals.
Role	A set of access control permissions within the system.
Saved Views	A menu to further specify the files that an Administrator or PowerUser may want to see, including Banned Files, New Pending Files, Malicious Files, Categorized Files, and Trusted Packages. Administrators or PowerUser also can Approve files (both globally and locally) and Ban files here.
SecCon (Security Condition)	The protection level applied to computers running Parity Client. A range of levels from Lockdown (most protective) to Agent Disabled (least protective) enable a Console User to specify the level of file blocking required. Additionally, there is an "offline seccon" which applies when the Parity Server cannot be reached.
Secure Sockets Layer (SSL)	Cryptographic protocol that provide security for communications over networks such as the Internet. SSL encrypts the segments of network connections at the Transport Layer end-to-end.
Simple Mail Transfer Protocol (SMTP)	Internet standard for electronic mail (e-mail) transmission across Internet Protocol (IP) networks. SMTP was first defined in RFC 821 (STD 15) (1982), and last updated by RFC 5321 (2008).
Simple Object Access Protocol (SOAP)	A protocol specification for exchanging structured information in the implementation of Web Services in computer networks.
Snapshot	A clean baseline for network files computer that Administrator or PowerUser have created for use in baseline drift analysis.
Software (File) Rules: Bans	Bans enable Console Users to specify files (by name or hash) to be blocked for particular groups of computers (i.e., by policy) or all computers at a site. Parity can ban files individually, and also can ban all files identified on a list of hashes the Console User provides.
Software Rules: Approvals	Several complementary software approval methods enable Console Users to approve legitimate software to run on all computers, on groups of computers (i.e., by policy) or to locally approve software to run on a single computer. Registry Rules Console Users can specify rules to protect specific registry key/value patterns from modification.
Transport Layer Security (TLS)	Cryptographic protocol that provide security for communications over networks such as the Internet. TLS encrypts the segments of network connections at the Transport Layer end-to-end.
Trusted Directory	When a Parity Agent contains a Trusted Directory, the contents (or changes) to that folder are sent from the client system to the Parity Console. The Parity Console then notifies the Parity Application Server Service. The Manifest Processing component of the Parity Application Server Service is responsible for validating the manifest file, importing it into the system, and then updating the client access policies appropriately.
User	A user can refer to either a Console User or a Client User.
Users	Console Users and Client Users
Whitelist	A list or collection of software or entities that are known, trusted, or explicitly permitted.

15 Bibliography

1. Common Criteria for Information Technology Security Evaluation – Part 1: Introduction and general model, Version 3.1 Revision 3.
2. Common Criteria for Information Technology Security Evaluation – Part 2: Security functional requirements, Version 3.1 Revision 3.

3. Common Criteria for Information Technology Security Evaluation – Part 3: Security assurance requirements, Version 3.1 Revision 3.
4. Common Evaluation Methodology for Information Technology Security Evaluation, Version 3.1 Revision 3.
5. Bit9, Inc. Parity™ 6.0.1 Security Target, Version 2.0, dated 22 February 2011
6. Evaluation Technical Report for a Target of Evaluation "Bit9, Inc. Parity™ 6.0.1" Evaluation Technical Report v3.0 dated 22 February 2011.