National Information Assurance Partnership

Common Criteria Evaluation and Validation Scheme Validation Report

Securify SecurVantage™,Version 3.1

Report Number: CCEVS-VR-04-0056
Dated: 26 January 2004
Version: **1.0**

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 6740
Fort George G. Meade, MD 20755-6740

1. Executive Summary

An evaluation of the Securify's SecurVantage™, version 3.1, was begun 21 February 2003 and completed 26 January 2004. Securify SecurVantage™ is a system that enables customers to define a network security policy (typically describing the permitted network operations), monitor networks for compliance with that policy, and produce relevant network operational information (such as events that fail to comply with the policy). The product consists of a tool for network security policy development and security analysis (Studio), a real-time monitoring system to continuously verify conformance to those security policies (Monitor), and an optional enterprise management and trend reporting system that can merge reports from multiple monitors (Enterprise).

The Target of Evaluation (TOE) includes those components developed by Securify, and not third-party components such as hardware and operating systems. The evaluation examined the threat of unauthorized users gaining control of the TOE, of attackers evading the monitoring implemented by the TOE, and of users raising their privileges in an unauthorized way. It is assumed that the TOE hardware and software will be located within controlled access facilities, preventing unauthorized physical access and protecting the TOE from unauthorized physical modification.

The evaluation was performed by CygnaCom in the United States. The evaluation was carried out in accordance with requirements drawn from the Common Criteria CCv2.1, Part 3 for EAL2 [CC_PART3] and Common Evaluation Methodology for Information Technology Security – Part 2: Evaluation Methodology [CEM_PART2]. The assurance activities in this level offer confidence that evaluated configuration of Securify SecurVantage™ (with documentation and software deliverables as defined in sections 6. and 8., respectively) contains requirements that are:

- Justifiably included to counter stated threats and meet realistic security objectives,
- Internally consistent and coherent
- Technically sound and
- Free from vulnerabilities associated with obvious and known threats.

CygnaCom, a Common Criteria Testing Laboratory [CCTL], is certified by the NIAP validation body for laboratory accreditation. The CCTL has presented CEM work units and rationale that are consistent with the CC, the CEM and CCEVS publication number 4 Guidance to CCEVS Approved Common Criteria Testing Laboratories [CCEVS_PUB 4]. The CCTL team concluded that the requirements of the EAL 2 have been met. Therefore, a **pass** verdict has been issued, by the CCTL, for Securify's SecurVantage™, version 3.1.

The information contained in this Validation Report is not an endorsement of Securify's SecurVantage™ by any agency of the U.S. Government and no warranty of the product is either expressed or implied.

1.1. Evaluation Highlights

Dates of Evaluation: 21 Feb 2003 – 26 January 2004
Evaluated Product: SecurVantage™, version 3.1
Developer: Securify Inc., 1157 San Antonio Road, Mountain View, CA 94043.
http://www.securify.com
CCTL: CygnaCom
Lead Evaluator: Nithya Rachamadugu (initially Kris Rogers)
Evaluation Class: EAL2
PPs Claimed: None.
Validation Team: David A. Wheeler, Institute for Defense Analyses (IDA)
Version of CC: Common Criteria version 2.1, August 1999
Version of CEM: Common Evaluation Methodology 1.0, August 1999

Effective Date for Interpretations: All interpretations as of 21 February 2003.

2. Product Identification

The Target of Evaluation (TOE) is Securify SecurVantage™, version 3.1. It consists of the following major components:

- Studio 3.1
- Monitor 3.1 or Monitor LE 3.1
- Enterprise 3.1 (optional)

The Monitor component can be configured in two different ways, depending on the processing speed required. High bandwidth solutions use Monitor 3.1, which has two subcomponents (Monitor SM and Monitor Harvester) each running on a different processor. For low bandwidth solutions, Monitor 3.1 LE is deployed, which uses the same software but uses a single hardware system that is shared between the two subcomponents.

The product must run on top of hardware and an operating system, all of which are part of the environment (and thus not evaluated). Monitor and Enterprise run atop Red Hat Linux 7.2 (patched) and an x86 system. Although it is also sold separately, in the evaluated configuration the hardware and operating system for Monitor and Enterprise are provided by the vendor directly to the customer, along with a CD-ROM permitting local re-installation. Studio is a user interface component and runs on Microsoft Windows 2000 on an x86 hardware platform. Additional interfacing uses a web browser, and the system depends on third-party encryption libraries, neither of which are part of the target of evaluation (TOE). See the Security Target (ST) for more information.

3. Security Policy

The TOE, with support from its IT environment, provides the following security functions (beyond self-protection):

- Auditing,
- Access Control,
- User Identification and Authentication
- Security Management

The primary purpose of the TOE is to permit administrators to define a network security (SecurVantage) policy, then monitor networks and report deviations from that policy. A SecurVantage Policy is a set of rules that describe the expected behavior of the systems within a network. Network objects represent systems. A network object can be one or many IP addresses. Each rule in the Policy describes how the system will log a network transaction between two network objects. All network transactions are logged and represented as an event. Each event represents the information contained in the headers of the actual packets within the network transaction. In SecurVantage, an output of the policy engine is created when network traffic is evaluated against a policy. A network event is a summary of the set of protocol events that make up a complete application level session on the network. For example, viewing a Web page creates a network event that summarizes the underlying IP association, TCP connection and HTTP Get protocol events.

The policy assigns by default a severity to every event, such that all events are logged by default. These default values can be changed by the user of the system to accommodate specific security policies. A severity is one of the following options: Critical, High, Medium, Warning, Monitor, Informational, or Ok. All events other than Ok are fully logged in the system down to the protocol details level (source and target network object name, ip addresses, protocols, src port, dst port, tcp flags, udp association, etc). Events that have a severity value of "ok" are only logged at a summary level (source and target network object name and service name). Events logged as critical are also called alerts and copied to a separate alert table. Alerts can trigger SMTP and SNMP messages to other management systems.

Effort was particularly expended to ensure that attackers could not easily inhibit or circumvent this monitoring. A given monitor can only implement one network security policy at a time. Multiple monitors may be part of a "domain", which shares a common network security policy. Enterprise (when used) can support multiple domains.

The TOE supports various user roles. Every user account is assigned one or more roles. The privileges granted to a user are the union of the privileges of that user's role(s). All roles (and their privileges) are simultaneously active for a given user. For example, a user with the

role "operator" and/or "analyst" (and no other role) cannot upload a new network security policy (i.e., cannot change the policy). The various roles, and their privileges, are shown below:

SecurVantage™ User Access Policy (from Security Target)

	Roles/Subjects					
Objects	Operator	Analyst	Developer	SV Manager	Account Manager	Super User
Event Data	View	View	View			View
Machines	View Status	View Status	View Status	View Status Start/Restart Stop Configure	View Status	View Status Start/Restart Stop Configure
DMEs		Download	Download			Download
User Access					Manage	Manage
Policy History	View	View	View			View
Policies		Extract	Upload Revert Extract			Upload Revert Extract
Alerts	Manage	Manage	Manage			Manage
Application Logs				View		View
User Logs					View	View

Details, including definitions of these objects, are given in the ST.

4. Assumptions and Clarification of Scope

4.1 Usage Assumptions

The following usage assumptions were made for the TOE.

It is assumed that administrators will have a strong understanding of the TOE, networking technology, and the network(s) they are monitoring. Fundamentally, the TOE allows an administrator to identify a policy of "normal" behavior, and the TOE will then report on all actions not corresponding to the policy. A knowledgeable administrator who creates a well-defined policy may find this TOE to be extremely effective at reporting just the events that need reporting. However, a poorly-defined policy (created by an administrator with insufficient understanding) may cause the TOE to report a voluminous number of unimportant events, and/or cause the TOE to omit events that were important to report. This

is not a fault of the TOE implementation; it is fundamental to the nature of its approach. Any TOE is best used by a knowledgeable administrator, but this TOE in particular requires a good administrator for effective use. Administrators should obtain training before use; the vendor makes such training available. Many deployments may want to ensure there are at least two trained administrators, to enable discussions of policy and to ensure continuous service if an administrator becomes unavailable. Administrator training and an understanding of the network being monitored are critical for effective and efficient use of this TOE.

At a more fundamental level, this TOE requires that it be possible to (eventually) determine the expected or permitted activities on the monitored networks, so that this information can be captured as a security policy. If all actions are permitted by all network components, the TOE's ability to compare actions with expected actions is far less valuable. A pre-existing written security policy, while very helpful, is not required; the security policy can be developed over time, starting with a more general policy and then repeatedly refining it. The TOE can also be used as a monitoring tool, so that actual network activity can guide formulation of the security policy. For nearly all real-life circumstances this is not a restriction. Most of today's networks *do* have a set of expected activities that is a small subset of all possible activities.

The TOE is normally used by plugging it into a switch's SPAN port. Typically SPAN ports only report the packets that *cross* the switch, and not network packets that appear on a network but do not cross it. This is an aspect of the environment, not the TOE itself; the TOE can only log what's reported to it. Since this is not an issue of the TOE itself, and many customers would expect this behavior anyway, it is not considered a vulnerability. However, administrators will need to configure their network and/or where they connect Monitor(s) so that what they wish to monitor can actually be monitored.

Users of the system are trusted with the privileges they have been granted, and it is presumed that authorized users will not misuse their privileges. For example, operators are trusted with the ability to view event data (which would give operators insight into all network activity) and analysts are granted the ability to download DMEs and extract policies (which would give analysts the ability to download summaries of network activities and know exactly what policy is being checked). However, as clearly noted above, users of the system are not fully trusted with all privileges. The TOE specifically works to prevent authorized users from gaining additional unauthorized privileges.

4.2 Environmental Assumptions

As stated in the ST, the hardware, operating system, and third-party cryptographic libraries are not included in the evaluation. As part of installation, the operating system used by Monitor and Enterprise (Red Hat Linux) is installed in a way that limits its functionality.

While this should help, for purposes of evaluation it is assumed that this operating system is secure in its environment. Note in particular that the operating system running Studio (Microsoft Windows) must be secured separately, and that no attempt is made by the product to ensure this. The TOE uses encryption to protect data between its major components, but since this encryption is performed in the environment (not in the TOE) it is not further considered here.

As clearly stated in the ST, the TOE only records IPv4 with normal Ethernet framing. Other kinds of data, particularly IPv6 and Ethernet jumbo frames (jumbograms), are not recorded. If all relevant network traffic is to be monitored, systems must be configured to reject these unrecorded packets. Many systems do this by default, so this is a plausible environmental restriction. If there are concerns that cooperating end-systems may surreptitiously send data between each other using other kinds of packets, then the network infrastructure should be configured to actively inhibit this kind of traffic. A particularly effective approach to doing this would be to insert network packet scrubbers that enforced these limits and regularized packets for monitoring purposes.

The TOE is capable of using the Domain Name Service (DNS) for translating IP addresses back to machine names. By default, this capability is disabled. Enabling this capability can aid administrators by giving them simple names instead of IP addresses. However, these name values are dependent on the security of DNS itself. Subversion of the DNS service could provide incorrect names. Also, attackers may control DNS services of other domains (legitimately or not). Thus, the names provided by DNS could be misleading. Note that the TOE does not use DNS for security decisions—this data is purely informational. The TOE's Studio component does include an ability to import DNS zone data from a file; ensuring that this zone data file is correct is outside the scope of the TOE. Administrators are warned about these issues in the installation guidance.

The TOE is capable of using the Network Time Protocol (NTP) for keeping time values correct. By default, this capability is disabled. NTP can be convenient for accurately keeping time values current. However, it is difficult to secure. An attacker that sends malicious NTP reports, or takes over a relevant NTP server, could cause the timestamps of events to be incorrect (impacting any Monitor report). This could also negatively impact attempts to merge data from multiple Monitors (as Enterprise does). Administrators are warned about this in the installation guidance (in the Administrator Addendum).

The TOE hardware and software must be located within controlled access facilities, preventing unauthorized physical access and protecting the TOE from unauthorized physical modification.

4.3 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 (EAL2 in this case).
2. This evaluation covers SecurVantage Version 3.1, not the later version 4.X series. Thus, these evaluation results do not automatically apply to version 4.X, and in particular additions in version 4.X (such as the Nessus security scanner) have not been considered by this evaluation.
3. As with all EAL2 evaluations, this evaluation did not specifically search for, nor seriously attempt to counter, vulnerabilities that were not "obvious" (as this term is defined in the CC and CEM) or "vulnerabilities" to objectives not claimed in the ST. A residual vulnerability (beyond the EAL2 level) *was* identified during the evaluation and was fixed by the developer.
4. In particular, the evaluation does not claim a resistance of the TOE to denial-of-service attacks, where an attacker intentionally causes a large load of traffic to mask different nefarious activities. This is a fundamental limitation of any passive monitoring tool. However, the TOE *does* log when its logging rates have been exceeded. Thus, although an attacker might be able to hide a nefarious action by overwhelming the TOE, the TOE *will* log when there was an opportunity to do so. Customers concerned about attackers who would use resource overwhelming attacks to hide other actions may consider purchasing the high-bandwidth Monitor, employing multiple Monitors, and/or employing network rate limiting components to slow network traffic to a monitorable rate.
5. Ideally, the monitoring system should log *exactly* what the receiving application would see. However, for a variety of reasons this is impossible for a passive monitoring system to do perfectly. In particular, when the TOE reassembles IP fragments, the TOE may reassemble them differently than the receiving system and thus report different results. This is a fundamental problem with all network monitoring systems, particularly passive monitoring systems ones like this. Since this case is a fundamental limitation of the technology, and would be expected by knowledgeable administrators anyway, this was considered acceptable. Customers who find this undesirable should counter this problem by inserting "network packet scrubbers" in their environment in such a way that the monitor and end-user receiving systems will see exactly the same data.
6. Network events can be a set of IP packets, not just one (e.g., initiating an HTTP request). If an event is halfway completed, but never actually completed, the TOE times out. However, the result of this time-out is logged, and the time-outs are longer than the standards required of receivers. This again raises the concern that receivers and the TOE may have a slightly different view of what is happening, but since there

is some logging no matter what the attacker does this was determined to be acceptable.

7. The purpose of the TOE is to report violations of policy, not identify network covert channels between cooperating parties. For example, two systems A and B could surreptitiously communicate with each other by communicating with a system C in accordance with a security policy, and then by monitoring each others' communication with C, extract the separate data intended for each other. Alternatively, systems A and B could arrange for data or processing at C to be an indirect channel between A and B. Steganographic tricks combined with error-correcting codes could make such hidden data particularly hard to identify. Detecting such channels is *not* the purpose of this TOE. The purpose of this TOE is to watch ordinary protocols and report direct violations according to a customer-defined security policy. Customers worried about these covert channel problems should consider redesigning their networks (e.g., to completely isolate the systems) or other measures.
8. The TOE does not log every byte of every packet involved. In many of its intended environments, this would be an extremely stressing requirement, would severely limit the length of time the logged information could be stored, and is unnecessary. Instead, the TOE stores a summary of every network event, with more information on events that are not ranked "ok." This information is sufficient for its intended purpose.
9. When a network event has a severity level of "critical," its information is copied to the separate "alerts" log. Note that operators and analysts (as well as developers and the super user) can manage the alerts log, i.e., they can both view and modify it. This is intentional. The expected use is that any of these roles can review the alerts, and once they have addressed them any of these users can remove the alerts that have been addressed. Removing an alert does *not* remove its original from the event data.
10. Network event data is stored at the Monitor that recorded it. This includes the criticality of the event, which is treated as a constant (it is assigned using the policy active at the time the event was recorded). Network event data can be retrieved as a DME file (if the user is permitted to do so); Studio can then be run locally to recompute network event severities using a different policy. These recomputed severities do not change the severity recorded by the relevant Monitor.

The ST does not claim conformance to any Protection Profile (PP), including the Intrusion Detection System (IDS) PPs. The TOE can be used for a variety of purposes, including monitoring network traffic to understand how the network is being used, or to enforce network policies having nothing to do with intrusion detection (e.g., to detect when authorized users of the network use network resources in a way that contravenes official policies). It is possible to configure the TOE to indirectly support intrusion detection, simply by defining a narrow policy of expected network usage. In this case, actions not in

accordance with the policy *might* indicate an intrusion. This is not always true; in particular, an intruder's actions might conform with the policy (depending on the policy and the intruder's actions). However, because of this general similarity, the IDS Sensor PP and the IDS Analyzer PP (both version 1.1, dated December 10, 2001) are worth comparing with this ST. Here is a brief comparison of the ST of this evaluation to the requirements of those PPs:

1. In general, the ST does not include in the TOE the hardware, operating system, and many 3rd party components (in particular, the encryption libraries). Instead, they are part of the environment. For example, the ST allocates encryption to the environment, as well as reliable time stamps. Meeting the PP would imply evaluation of many other components (by enlarging the definition of the TOE).
2. The ST does not include an inter-TSF availability within a defined availability metric, nor inter-TSF detection of modification (FPT_ITA.1, FPT_ITI.1).
3. The ST has fewer auditing requirements on TOE-specific actions (see FAU_GEN), i.e., reading of information from the audit records is not specifically required in the ST.
4. The ST does not require authentication failure handling (FIA_AFL.1).

Note that this TOE is intended for a different purpose than an IDS. This TOE's purpose is to detect violations of a security policy, not specifically to detect intruders. Nevertheless, the TOE is sufficiently flexible to support many kinds of intrusion detection if desired. Users who intend to use the TOE this way can use the above comparison to see if these differences are acceptable to them. Note that the TOE can also be used in parallel with a system specifically designed for intrusion detection; the IDS system can detect certain kinds of intrusions, while the TOE could detect other actions that violate policies (such as certain misuses of network resources by authorized users) which may also catch intruders that slip by the IDS.

The ST provides additional information on the assumptions made and the threats countered.

5. Architectural Information

SecurVantage™ consists of three major components:

1. SecurVantage™ Studio: provides a management interface that allows for the authoring of network security policy at multiple levels.
2. SecurVantage™ Monitor: captures and evaluates monitored network traffic according to the security policy
3. SecurVantage™ Enterprise: combines the information from multiple monitoring points into a single, real-time monitoring and management console.

Users use Studio to define the network security policy using a proprietary policy language. This security policy defines the "correct" behavior of the network(s) being monitored.

Users may use a web browser to communicate with Monitor or Enterprise over an encrypted (SSL) link. When initially connecting to Monitor or Enterprise, users are presented with a new self-signed server certificate, which they can verify by comparing these certificates to the values generated during Monitor/Enterprise installation. Users are first authenticated (using username and password sent over the encrypted link), and depending on their roles users may change data (such as uploading a new policy defined using Studio) as well as receive data (such as event data). Note that the web browser and third-party encryption libraries are outside the TOE.

Studio also supports query-only direct access to Monitor or Enterprise. When using Studio in this way, Studio also uses an encrypted SSL link (to the same SSL/HTTP port). This access approach supports the username/password pair (it is the same mechanism as above), but in addition it also permits users to set up client-side certificates. If client-side certificates are set up, users need not separately log into Monitor or Enterprise simply to query data (the client-side certificates are instead protected by the operating system Studio runs on). Note that this direct access only permits query operations, so only users with the role of operator, analyst, developer, or super user can usefully use this access approach. This access approach cannot be used to modify information, in particular it cannot be used to modify policies.

SecurVantage™ Monitor captures and evaluates in real time the packets flowing through the network at all levels of the protocol stack. It then makes decisions on whether the traffic is consistent with the policy specification. The result is a set of "network events" with each event including an attribute termed "criticality" in the ST. Monitor has two subcomponents:

1. Monitor Harvester, which obtains network traffic. It contains the Monitor Securify Packet Filter Module (SPFM), which actually captures the traffic.
2. Monitor Security Master (SM), which compares the harvested network traffic with the customer-defined policy. This contains the Security Policy Module (SPM), which actually compares the traffic to the policy.

Monitor LE runs both subcomponents on a single processor, which lowers hardware costs but also lowers performance. Monitor (not LE) runs the components on two separate processors, with a completely private network connection between them.

If an Enterprise system is deployed, Enterprise copies information from the Monitors connected to Enterprise and aggregates them into a local database. This database is accessible through the web interface for a period of 48 hours. The Enterprise serves also as a conduit to the Monitors' databases when detailed information is requested by Studio application. Enterprise can also deploy policies to multiple Monitors.

When Enterprise is deployed, Enterprise and the Monitor(s) communicate using SSL. For this communication, certificates are initially verified the first time the Enterprise and Monitor components communicate. Both server and client certificates are used.

Figure 1 shows a typical deployment of SecurVantage™, although SecurVantage™ Monitor can be placed anywhere on the network. It does not necessarily have to be on its own sub-network and does not have to be connected through a switch. Typically SecurVantage™ Monitor is connected to the SPAN port of a switch where there is traffic relevant to the policy. However, there are no assumptions about the source of the traffic.

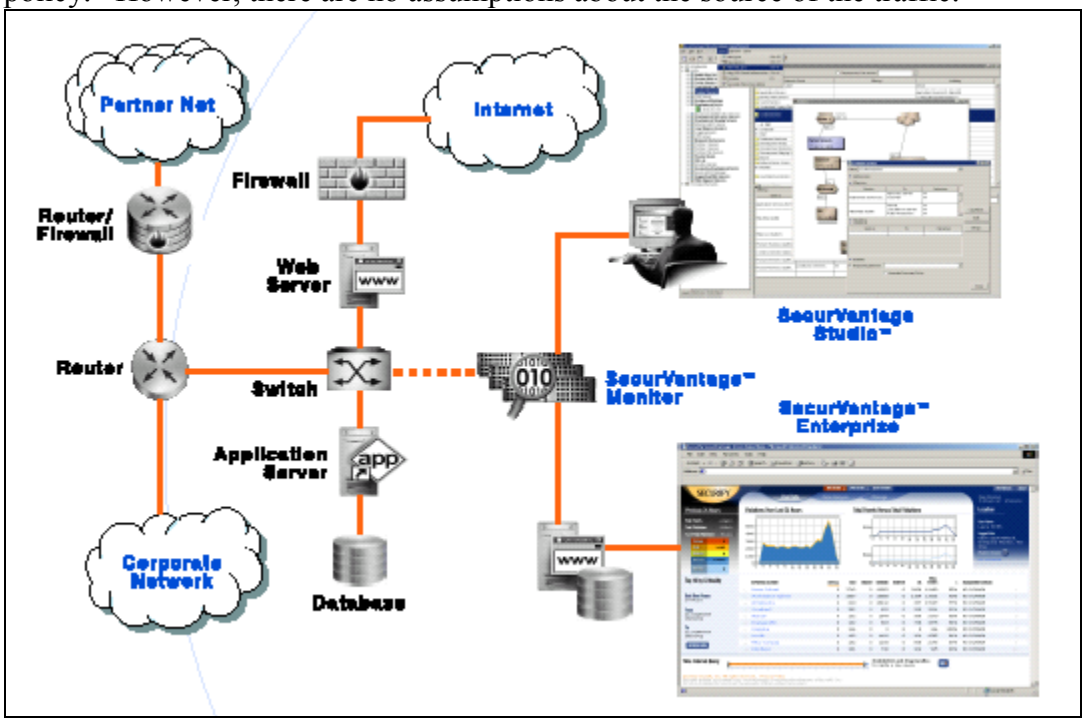

Figure 1: Typical SecurVantageTM Deployment

SecurVantage™ consists of the policy development and analysis environment coupled with the monitoring system and the enterprise management system. Figure 2 shows the System Architecture.

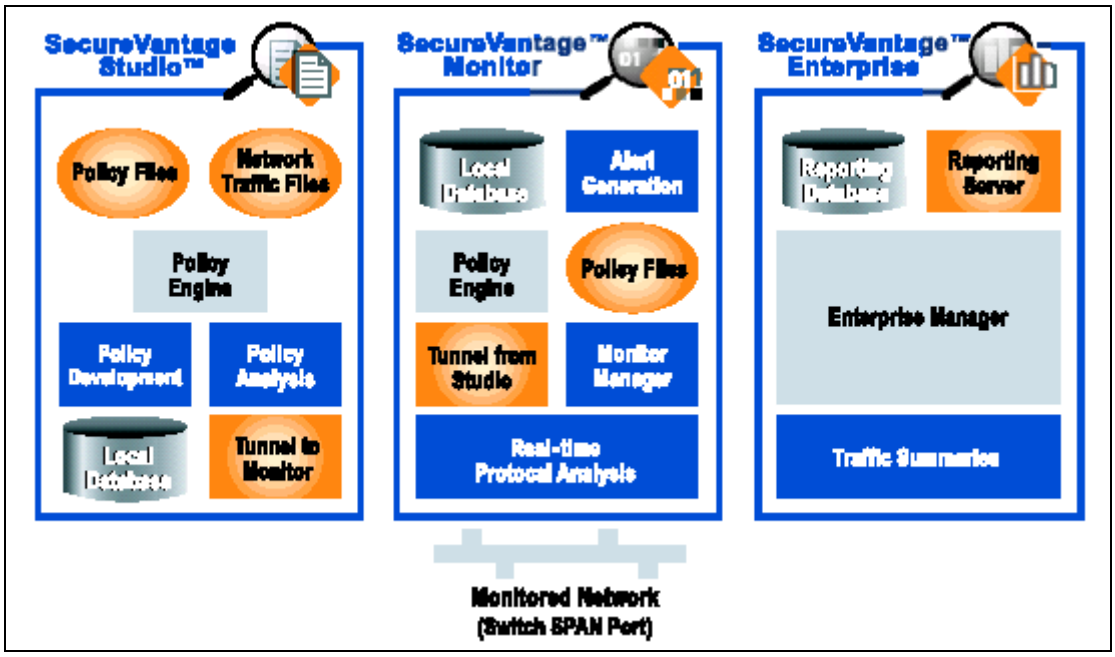

Figure 2: SecurVantageTM System Architecture

See the Security Target section 2.2 for additional discussion of the TOE's architecture, including more information on the tasks performed by each major component.

6. Documentation

The documentation provided with the product to customers is as follows:

- Securify SecurVantage 3.1 Deployment Guide. November 2002.
- Securify SecurVantage 3.1 Installation Guide. November 2002.
- Securify SecurVantage 3.1 Operations Guide. November 2002.
- Securify SecurVantage 3.1 Release Notes. November 2002.
- Securify SecurVantage 3.1 Common Criteria Addendum. January 7, 2004
- Securify SecurVantage 3.1 Administrator Addendum. January 21, 2004. This document complements the information contained in the SecurVantage 3.1 Operations Guide and the SecurVantage Common Criteria Addendum.

For a longer list of the major pieces of evidence examined during the evaluation, see section 14 of this report.

7. IT Product Testing

7.1 Examination of Vendor Tests

The vendor provided test plans, procedures, test results and a test coverage document. The security testing was driven by the SecurVantage "Test Matrix," a spreadsheet divided into three sections: (1) IT Security Requirements (as section 6 in the ST), (2) tests for Delivery, Installation and Configuration, and (3) other tests (that test important general functionality of the TOE instead of a specific requirement). The IT Security Requirements section (the first section) is further divided into three subsections: Studio, Monitor and Enterprise. These are indexed by the actual requirements in the ST's TSS.

Each test was performed by one of the following "test tools":

1. Test Matrix: This is a document (supporting the spreadsheet) that specifies step by step how to perform a specific test. The document is indexed by a test case number.
2. Sentinel: This is an automated tool for testing web applications. The test case is usually mapped to a signature of the sentinel test.
3. By Inspection: Guidance necessary to perform the test.
4. By code inspection: Tests that can only be verified through a source code inspection

For purposes of testing, a special test harness was used. Normally, the system is connected into a switch's SPAN port. However, testing the system by directly connecting to a SPAN port and then generating data from multiple different networks would result in repeated tests not generating identical inputs to the TOE (due to different interleaving). Thus, for testing purposes, instead of connecting to a switch's SPAN port, the TOE was directly connected to systems which replay data previously captured from a SPAN port. The evaluator and validator examined this configuration and were satisfied that the test harness produced the same inputs (as seen by the TOE) as the original network whose traffic had been captured (including Ethernet MAC addresses).

The evaluator determined that the vendor tested (at a high level) most security-relevant aspects of the product. The evaluator determined that the developer's tests were sound in their approach. The test document provided the configuration of the test hardware and software, the objective for each of the tests, and test procedures. The information provided was adequate to be able to reproduce the tests. The evaluators determined that the developer's approach to testing the TSF was appropriate for this EAL2 evaluation.

7.2 Evaluator Independent Tests

The evaluator performed the tests at the developer's site using the equipment provided by the developer. The tests were performed in two configurations; one configuration representing larger installations of the TOE which contained an Enterprise manager managing multiple (in this case two) domains and containing both (Monitor and LE) versions of the Monitor and one Studio. The second configuration represented a smaller implementation consisting of only a Studio and the Monitor LE. The enterprise and Studio were connected via separate network to the Monitor. Though this is not an imposed restriction, this is the most common field installation configuration. The same hardware was used for both configurations. The evaluator performed installation and testing on one configuration and then reconfigured and repeated the tests on the other configuration.

The evaluator installed the TOE using the installation procedures. About 90% of the developer tests were repeated. The evaluator used the developer's automated tool Sentinel for some of the tests. The evaluator repeated some tests manually to gain confidence in the tool. About 80% of the developer's manual tests were also repeated. The developer provided a traffic generator tool that replayed the traffic from a previous session. This provided a varied sample of the traffic and helped to regulate the speed of the traffic as desired.

Functions that were deemed critical to the operation of the TOE components (for example. DME creation, policy pushing, super-operator functions, creation and revoking of users, critical event viewing), difficult tests and unusual actions (like the rate limiting feature and data availability (database roll-over)), and the normal operation scenario were chosen as the basis for the evaluator's tests.

Critical messages were generated and the SNMP, SMTP message generation was verified for critical alerts. For the test repeated from the developer suites, the evaluator examined the test results and found them to be matching those of the developer. Any mismatches were purely due to data related inconsistencies. The overall verdict of the evaluator testing is that the TOE components perform the security functions in accordance with those specified in the ST and the developer's test results match those of the evaluators.

7.3 Strength of Function

Even if a TOE security function cannot be bypassed, deactivated, or corrupted, it may still be possible to defeat it because there is a vulnerability in the concept of its underlying security mechanisms. For those functions a qualification of their security behavior can be made using the results of a quantitative or statistical analysis of the security behavior of these mechanisms and the effort required to overcome them. The qualification is made in the form of the strength of TOE security function claim.

The security of this TOE depends on the strength of the passwords used to access Monitor and Enterprise. The original system analysis in the evaluation identified some concerns, because the web interface had a lax password policy ("at least 6 characters" with no other requirements). The vendor decided to modify the product to require stronger passwords. The TOE now enforces a safer (more stringent) password policy in its web (SSL) interface:

- Minimum of 8 characters in the password (maximum 64 characters).
- At least one lower case character.
- At least one upper case character
- At least one numeric character.

The TOE enforces a requirement that console access passwords be at least 8 characters long.

For both web interface and console access, the administrator guidance (given in the Administrator Addendum) requires that passwords for both the web interface and console meet these requirements:

- Minimum of 8 characters in the password.
- At least one lower case character.
- At least one upper case character.
- At least one numeric characters.
- At least one printable character that is non-alphanumeric.
- Cannot be a single dictionary word with non alphabetic characters appended and/or pre-pended.

A strength-of-function analysis took these password requirements (as well as other information) and justified a ranking of SOF-basic. The overall SOF requirement for the TOE made in ST is expressed as an SOF rating, SOF-basic. Thus, the TOE meets the ST requirements.

7.4 Vulnerability Analysis

Vulnerability analysis is a process for identifying potential vulnerabilities and determining whether potential vulnerabilities identified throughout the evaluation process could allow users to violate the TSP. See the CC and CEM for additional information on the requirements for an EAL2 evaluation.

The vendor performed a vulnerability analysis in three parts, searching for:

1. Specific publicly-known vulnerabilities in the product (including third-party components)
2. Obvious vulnerabilities in the Enterprise and Monitor web interfaces.
3. Obvious vulnerabilities in the process of collecting, storing and presenting network traffic.

For the latter two areas, the obvious vulnerabilities searched for were a set of well-known, publicly described attacks, such as buffer overflows.

The vendor searched for publicly known vulnerabilities specifically related to SecurVantage, as well as publicly known vulnerabilities in the third-party products used by SecurVantage (and are in the environment of the evaluation). This search included searches in the information of Security Focus (www.securityfocus.com), Bugtraq (through Security Focus), Packetstorm (www.packetstormsecurity.org) and CVE (cve.mitre.org). No publicly-known vulnerabilities specific to SecurVantage were found. The known vulnerabilities in the third-party products were examined, and were either countered or shown to be unlikely to be a vulnerability in this circumstance.

Searches for vulnerabilities in the web UI concentrated on these topics:

1. Parameter Verification. All user entry fields in the web UI to Monitor and Enterprise were tested for SQL injection, string manipulation, path traversal and buffer overflow type of attacks. Some of these attacks were performed via automatic tools like Spike or Securify's proprietary Sentinel.
    a. SQL Injection. Parameters throughout the application were tested for SQL injection. The approach was to follow usual SQL injection attacks to try to stop the SQL command issued and concatenate a new SQL statement to be issued; or via SQL statements try to eliminate the constraints of the original SQL request. In every case the application returned an application level error and there was no indication that it would be susceptible to SQL attacks. Although not required by an EAL2 evaluation, the vendor also performed a selected code inspection of pieces of the code they thought were likely to be stressed and confirmed that the code properly validated the input strings.
    b. String Manipulation. The vendor attacked the web interfaces by adding special characters, or change expected string values for numeric values, different types of encoding, and so on.
    c. Path Traversal: Upload/Download. There are features of the UI that would allow an user to upload and download files from the system. The vendor attempted to download files other than the ones specified in the application without any success. The vendor further verified that the application has a very strict chroot environment and permissions were properly set throughout the file system, which enforced these restrictions. The vendor also attempted to upload files with names that would attempt some form of path traversal in an attempt of corrupting valid files in the system. The software upgrade feature of the system implements a digital signature, creating a further barrier. In short, invalid activities were prevented by the protection mechanism in the server.

   d. Buffer Overflow. The vendor tested many parameters for very large inputs from the user. Most of these tests were conducted using Spike.
2. User Role Model (malicious URLs). The vendor stressed the role model by attempting both breaking the assumptions in the provided UI and the model itself. The vendor attempted to bypass the user-role model by crafting specific requests (URLs) to the application. The user role model was correctly enforced by the system, as the server before performing any action would check the permissions on the credential making the request.
3. Corrupted Certificates.

Vulnerabilities in the ability of the system to log and display network events emphasized these topics:

1. Rate Limiting. If the Monitor is configured to monitor a network where a specific transaction occurs at a high rate and this transaction is deemed critical by the policy it could slow the collection of other type of violation with lower level of criticality. The system was designed under the assumption that critical violations will always get properly logged and would have priority over other type of violations. So this behavior is consistent with the design of the system. An attacker could attempt to overwhelm the system, but that attempt would be clearly visible.

   The monitor implements rate limiting when the amount of data coming into the monitor is excessive. An attacker could decrease the effectiveness of the monitor by generating excessive traffic that is permitted by policy, causing the monitor to start to drop data. Theoretically, the monitor might not register a subsequent attack. The monitor software acts to prevent this possibility as follows:
   a. When possible, the monitor discards only data that is redundant with other data already collecting (e.g., drops the 1,000-20,000th instances of the same kind of event, while still collecting the 1-100th instances of a new kind of event).
   b. When the above is not possible, the monitor implements randomization algorithms to make it difficult for such an attacker to predict which events will be dropped, so it is hard for an attacker to guarantee that an attack goes unnoticed.
2. Buffer Overflow. The following cases were taken into consideration while performing the buffer overflow analysis to the data collection ability of the SecurVantage Monitor.
   a. Ethernet Jumbo Frames. Jumbo frames can theoretically overrun buffers when programs assume that all frames will fit into the Ethernet frame size of 1518 bytes any copy packets into such a buffer without first checking its size. Monitor does not support jumbo frames. Such frames are dropped without policy evaluation, and thus cannot be used to attack the monitoring itself. The

ST was clarified to clearly state that such frames are not recorded; see Jumbo Ethernet frames section below.

b. Oversized IP Datagram Reassembly. The evaluator, working with the vendor, identified the need to ensure that oversized IP datagrams would not be a serious vulnerability. By manipulation of IP fragment fields, it is possible to construct from several frames a single IP datagram that is larger than the maximum allowed size of 64Kbytes. (This is the most obvious way of attempting to create a buffer overflow at the IP level). Monitor does not crash when presented with such datagrams. When such a datagram is encountered, it is reassembled correctly (the length field in the IP header will be incorrect due to overflow of the 16 bit value). Monitor will not crash or cause other untoward effects when this occurs.

c. Long Data in Protocols. Many protocols allow variable length data. Examples are URLs in HTTP requests, filenames in an FTP session and so on. When processing such data the monitor must avoid buffer overflows or resource exhaustion. Not only must such data not overflow buffers, but SecurVantage™ must also avoid attempting to log infinitely large amounts of data to the database. So URLs, etc., must be truncated at some reasonable value. The vendor captured sessions where such extremely long values occur, in particular HTTP sessions with very long URLs and FTP sessions with very long filenames. The vendor replayed the captured sessions and verified that the Monitor would correctly capture and store these sessions. The vendor also verified that they were correctly displayed. By code inspection the vendor verified that information was truncated even prior of being stored in the database, so the risk of overflow in the database was handled as well. ICMP packets up to the maximum IP datagram size were crafted, and it was verified that Monitor logged them correctly.

3. DNS. Monitor can be configured to connect to a DNS server to resolve names to IP addresses. Resolved names are displayed in the web UI whenever a user positions the mouse cursor at an IP address in the data analysis pages. The vendor performed a code review on the DNS client implemented in the Monitor to assess a potential vulnerability. The assumption for this analysis is than attacker compromises the DNS server configured in the environment, or legitimately controls a DNS server (of a domain the attacker owns), and can send arbitrary DNS responses to a specific DNS query. The vendor considered the following cases: buffer overflows, abuse of DNS pointers, SQL injection, corruption of user interface, and crashing of the system by unexpected input. No way was found to use DNS to remotely interfere with the TOE security requirements. However, note that DNS data may be provided by an attacker, and thus the data may not have the value expected by users. Thus, guidance was added to suggest leaving the DNS access capability disabled, and warning about this issue if DNS access is turned on.

4. Other techniques to avoid monitoring.

One area of concern eventually resolved during the evaluation was that Monitor drops Ethernet jumbo frames. There was concern that an attacker could take advantage of this deficiency to try to exploit systems in the network, without being logged by Monitor, using Ethernet jumbo frames. This was countered by asserting that the environment does not accept Ethernet jumbo frames. However, the U.S. CCEVS does not accept STs with nonexistent or completely unrealistic environments, so the vendor had to briefly demonstrate that this specified environment was reasonable. The vendor showed that there were plausible infrastructure components (Cisco 4xxx, 5xxx, and 6xxx switches) and end-user systems (Windows 2000, Windows XP, Red Hat Linux 7.2 running Linux kernel 2.4, Sun Solaris) that by default drop any Ethernet jumbo frames they receive. Thus, this environmental requirement appears reasonable, because there are reasonable environments where this assumption is true.

The evaluator determined that the product met the criteria of EAL2 for vulnerability analysis.

8. Evaluated Configuration

The evaluated configuration was configured per the documents listed in section 6 of this report. The system was tested with DNS and NTP services disabled, as is its default (see section 4.2 for a discussion on these services). For additional information on how to securely deploy this TOE (beyond the referenced documentation), see sections 4 and 10 of this report.

9. Results of the Evaluation

The validation team followed the procedures outlined in the Common Criteria Evaluation Scheme [CCEVS] publication number 3 for Technical Oversight and Validation Procedures. [CCEVS_PUB 3]. The validation team has observed that the evaluation and all of its activities were in accordance with the Common Criteria, the Common Evaluation Methodology [CEM], and the CCEVS. The validation team therefore concludes that the evaluation and its results of **pass** are complete and should be approved by the CCEVS.

9.1 Assurance Content

The evaluation provides for Assurance at the EAL 2 level without augmentation. Therefore, this includes the assurance components as shown in the table below:

EAL2 Assurance Requirements

Assurance Class	Assurance Family
ST Evaluation	ASE_DES.1

Assurance Class	Assurance Family
	ASE_ENV.1
	ASE_INT.1
	ASE_OBJ.1
	ASE_PPC.1
	ASE_REQ.1
	ASE_SRE.1
	ASE_TSS.1
Configuration Management	ACM_CAP.2
Delivery and Operation	ADO_DEL.1
	ADO_IGS.1
Development	ADV_FSP.1
	ADV_HLD.1
	ADV_RCR.1
Guidance Documents	AGD_ADM.1
	AGD_USR.1
Tests	ATE_COV.1
	ATE_FUN.1
	ATE_IND.2
Vulnerability Assessment	AVA_SOF.1
	AVA_VLA.1

10.Validator Comments/Recommendations

As with any evaluation, this evaluation shows that the evaluated configuration meets the security claims made, with a certain level of assurance.

Be sure to note the assumptions and clarifications of scope in section 4 of this report. In particular:

1. As clearly stated in the ST, the TOE only records IPv4 with normal Ethernet framing. Other kinds of data, particularly IPv6 and Ethernet jumbo frames (jumbograms), are not recorded. If all relevant network traffic is to be monitored, systems must be configured to reject these unrecorded packets. Many systems do this by default, so at this time this is a plausible environmental restriction for many circumstances.

2. Note that this tool is designed to report violations from a customer-defined network security policy. Thus, it is strongly advised that administrators be trained so that they can devise a good network security policy for their environment.
3. The tool is not primarily intended to be an intrusion detection tool, but it can be especially effective in detecting certain kinds of intrusions (see section 4.3 for a discussion of this).

For many purposes, the TOE is useful as it is. However:

1. Consider connecting Enterprise (if present), Studio, and all Monitor(s) using a separate private ("command and control") network solely allocated for this purpose. The evaluated configuration presumed that the network connecting the TOE components was accessible by an attacker, and examined the TOE resistance to attack in that circumstance. No EAL2-level vulnerabilities were found. However, using a private network can reduce even further the opportunities for an attacker to exploit the TOE or to assail it using denial-of-service attacks. A physically separate network would be even better than a logically separated one.
2. Firewalls may be useful to prohibit certain actions that should be simply prohibited. This would reduce the processing load on the TOE so it can use its processing power to monitor the subtler activity the TOE is capable of monitoring.
3. It may be prudent to pay special attention to the workstation(s) used to run Studio: lock down its operating system to be secure, rigorously maintain its operating system for patches, use it only for SecurVantage-related work (and isolate it administratively for just that purpose), and power down or unplug the workstation when it is not in use. TOE users with lower privileges should not have special privileges to the workstations' operating system used by TOE users with higher privileges, since those special privileges could be exploited to gain control over the other TOE user.
4. Some high-risk environments that decide to use this EAL2 evaluated TOE may also find it useful to augment this TOE with other tools, since no one tool has all strengths and some tools are especially good complements for this TOE. One useful type of complementary tool would be a network scrubber (with rate limiting), to ensure that the end-systems and the Monitor see exactly the same data and that the data rate is loggable. Another such tool would be specialized intrusion detection systems (IDSs). IDSs could detect attack attempts (even if the communication is allowed by the TOE security policy), while the TOE can detect misuse of the network (including certain kinds of intruder activity) in a way that many IDSs would not detect.
5. Be careful when updating the security policy, since this interferes with capturing network events when the policy is being updated. Avoid updating the security policy at predictable times, and if collecting all network events is critical, consider temporarily disabling the network for a period of time while updating the security policy. As long as security polices are not updated at times known by an attacker, this is more of a theoretical problem than a real one.

6. The safest course is to leave NTP and DNS access disabled. DNS can be enabled, but it's important for users to understand that in some cases an attacker may provide the displayed DNS data. See section 4.2 for a discussion of these issues.

The tool can detect and report violations of a security policy, but what happens after detection is a decision humans must make. If the goal is to detect authorized users performing certain prohibited actions, it will be important to ensure that users know (in a general way) what kinds of actions are allowed and what is prohibited. It may be necessary to develop a more detailed human-readable network policy so that authorized users will know what actions they may and may not take.

The validator observed that the evaluation and all of its activities were in accordance with the CC the CEM, and CCEVS practices; and that the CCTL presented appropriate CEM work units and rationale. The validation team therefore concludes that the evaluation, and its results of **pass,** are complete, correct, and should be approved by the CCEVS.

11. Annexes

None.

12. Security Target

The Security Target is provided separately. It is Version 2.0, dated January 26, 2004.

13. Glossary

The following acronyms are provided for reference:

CC	Common Criteria
CCEL	Common Criteria Evaluation Laboratory
CCEVS	Common Criteria Evaluation and Validation Scheme
CCTL	Common Evaluation Testing Laboratory
CEM	Common Evaluation Methodology
CI	Configuration Items
CSC	Computer Sciences Corporation
DSA	Developer Security Analyst
EAL	Evaluation Assurance Level
EDR	Evaluation Discovery Report
ETR	Evaluation Technical Report
MRA	Mutual Recognition Arrangement
NIAP	National Information Assurance Program
NIST	National Institute of Science & Technology
NSA	National Security Agency

OR	Observation Report
PP	Protection Profile
SAR	Security Assurance Requirement
SFR	Security Functional Requirements
SOF	Strength of Function
ST	Security Target
TCSEC	Trusted Computer Systems Evaluation Criteria
TOE	Target of Evaluation
TSC	TOE Scope of Control
TSF	TOE Security Functions
TSFI	TSF Interface

The following CC terms are provided for reference:

User	Any entity (human user or external IT entity) outside the TOE that interacts with the TOE.
Human user	Any person who interacts with the TOE.
Authorized User	A user that, in accordance with the TOE Security Policy (TSP) may perform an action. (As identified by group membership.)
External IT entity	Any IT product or system, untrusted or trusted, outside of the TOE that interacts with the TOE.
Role	A predefined set of rules establishing the allowed interactions between a user and the TOE.
Identity	A representation (e.g., a string) uniquely identifying an authorized user, which can be either the full or abbreviated name of that user or a pseudonym.
Authentication data	Information used to verify the claimed identity of a user.

14. Bibliography

The evidence used in this evaluation is based upon the product, the security target, the user documentation (see section 6 for list), and the following documentation:

1. Performance Measures for SecurVantage, November 10, 2003
2. Securify SecurVantage 3.1 Auditing Matrix, December 2003

3. Securify SecurVantage 3.1 CLI, December 2003, December 29, 2003
4. Securify SecurVantage 3.1 Configuration Mechanisms and Description, January 21, 2004
5. Securify SecurVantage 3.1 Configuration Parameters, December 16, 2003
6. Securify SecurVantage 3.1 Data Purging Specification, September 24, 2002
7. Securify SecurVantage 3.1 EM Coversheet, December 16, 2003 (EM stands for "Enterprise Manager")
8. Securify SecurVantage 3.1 External Interfaces, January 7, 2004
9. Securify SecurVantage 3.1 High-Level Design (HLD), January 7, 2004
10. Securify SecurVantage 3.1 Manufacturing Procedures, December 16, 2003
11. Securify SecurVantage 3.1 Monitor Coversheet, January 7, 2004
12. Securify SecurVantage 3.1 Monitor Database Schema, April 22, 2002
13. Securify SecurVantage 3.1 Monitor LE Coversheet, December 16, 2003
14. Securify SecurVantage 3.1 Procedures to audit capture and storage of network events,December 16, 2003
15. Securify SecurVantage 3.1 RCR, January 7, 2004
16. Securify SecurVantage 3.1 Strength of Function Computation, December 11, 2003
17. Securify SecurVantage 3.1 Test Matrix, December 29, 2003
18. Securify SecurVantage 3.1_80.cvstags, December 16, 2003
19. Securify SecurVantage 3.1Physical Data Model, December 16, 2003
20. Securify SecurVantage Vulnerability Analysis, version 1.8, January 21, 2004
21. Test2.zip (Collection of Test documentation), October 24, 2004
22. Test3.zip (Supplemental collection of test documentation), October 29, 2004
23. Testing TCP Checksums, January 23, 2004

The evaluation and validation methodology was drawn from the following:

[CC_PART1]	Common Criteria for Information Technology Security Evaluation – Part 1: Introduction and general model, dated August 1999, version 2.1.
[CC_PART2]	Common Criteria for Information Technology Security Evaluation – Part 2: Security functional requirements, dated August 1999, version 2.1.
[CC_PART2A]	Common Criteria for Information Technology Security Evaluation – Part 2: Annexes, dated August 1999, version 2.1.
[CC_PART3]	Common Criteria for Information Technology Security Evaluation – Part 3: Security assurance requirements, dated August 1999, version 2.1.

[CEM_PART1] Common Evaluation Methodology for Information Technology Security – Part 1: Introduction and general model, dated 1 November 1997, version 0.6.

[CEM_PART2] Common Evaluation Methodology for Information Technology Security – Part 2: Evaluation Methodology, dated August 1999, version 1.0.

[CCEVS_PUB 1] Common Criteria, Evaluation and Validation Scheme for Information Technology Security, Organization, Management and Concept of Operations, Scheme Publication #1, Version 2.0, May 1999.

[CCEVS_PUB 2] Common Criteria, Evaluation and Validation Scheme for Information Technology Security, Validation Body Standard Operating Procedures, Scheme Publication #2, Version 1.5, May 2000

[CCEVS_PUB 3] Common Criteria, Evaluation and Validation Scheme for Information Technology Security, Technical Oversight and Validation Procedures, Scheme Publication #3, Version 1.0, January 2002.

[CCEVS_PUB 4] Common Criteria, Evaluation and Validation Scheme for Information Technology Security, Guidance to CCEVS Approved Common Criteria Testing Laboratories, Scheme Publication #4, Version 1, March 20, 2001

[CCEVS_PUB 5] Common Criteria, Evaluation and Validation Scheme for Information Technology Security, Guidance to Sponsors of IT Security Evaluations, Scheme Publication #5, Version 1.0, 31 August 2000.