National Information Assurance Partnership

Common Criteria Evaluation and Validation Scheme
Validation Report

Finjan Software Incorporated
SurfinGate Version 5.6

Report Number: CCEVS-VR-01-0006

Dated: October 31, 2001

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

ACKNOWLEDGEMENTS

Validation Team
Jerome Myers
Aerospace Corporation
Columbia, Maryland

Manilal Daya
Yi-Fang Koh
The MITRE Corporation
Bedford, Massachusetts

Common Criteria Testing Laboratory

Science Applications International Corporation
Columbia, Maryland

National Information Assurance Partnership

Common Criteria Certificate

Finjan Software, Incorporated

The IT product identified in this certificate has been evaluated at an accredited testing laboratory using the Common Methodology for IT Security Evaluation (Version 1.0) for conformance to the Common Criteria for IT Security Evaluation (Version 2.1). This certificate applies only to the specific version and release of the product in its evaluated configuration. The product's functional and assurance security specifications are contained in its security target. The evaluation has been conducted in accordance with the provisions of the NIAP Common Criteria Evaluation and Validation Scheme and the conclusions of the testing laboratory in the evaluation technical report are consistent with the evidence adduced. This certificate is not an endorsement of the IT product by any agency of the U.S. Government and no warranty of the IT product is either expressed or implied.

Product Name: Finjan SurfinGate
Version and Release Numbers: Version 5.6
Evaluation Platform: Option 1: Windows NT 4.0 Service Pack 4 or later or Windows 2000 for all SurfinGate Components
Option 2: SurfinGate Server and Database on Sun Solaris Version 2.6 with Oracle Version 7.3.4 and with SurfinConsole on Windows NT 4.0 Service Pack 4 or later or Windows 2000

Name of CCTL: Science Applications International Corporation
Validation Report Number: CCEVS-VR-01-0006
Date Issued: 31 October 2001
Protection Profile Identifier: N/A
Assurance Level: EAL3

Original signed

Director
Information Technology Laboratory
National Institute of Standards and Technology

Original signed

Information Assurance
Director
National Security Agency

Table of Contents

LIST OF FIGURES

LIST OF TABLES

1 EXECUTIVE SUMMARY

The Finjan SurfinGate, Version 5.6 has been evaluated at an accredited testing laboratory using the Common Methodology for IT security Evaluation (Version 1.0) for conformance to the Common Criteria for IT Security Evaluation (Version 2.1). This Validation Report applies only to the specific version and release of the product in its evaluated configuration. The product's functional and assurance security specifications are contained in its security target. The evaluation has been conducted in accordance with the provisions of the NIAP Common Criteria Evaluation and Validation Scheme and the conclusions of the testing laboratory in the evaluation technical report are consistent with the evidence adduced. This Validation Report is not an endorsement of the IT product by any agency of the U.S. Government and no warranty of the IT product is either expressed or implied.

This report documents the NIAP validators' assessment of the CCEVS evaluation of Finjan SurfinGate Version 5.6. It presents the evaluation results, their justifications, and the conformance result.

The evaluation was performed by Science Applications International Corporation (SAIC) and was completed on October 29, 2001. The information in this report is largely derived from the Evaluation Technical Report (ETR) written by SAIC and submitted to the validators. The evaluation determined the product to be **Part 2 conformant, Part 3 conformant,** and to meet the requirements of **EAL 3**, resulting in a "pass" in accordance with CC Part 1 paragraph 175.

SurfinGate is a malicious code detection and response tool. The product provides network administrators with a vehicle for managing the risk of introduction of malicious code through the HTTP (web) interface. It provides services to collect web-based traffic, analyze the collected web-based traffic against an administrator-defined policy, and respond to any security violations detected. The analysis of web traffic and blocking of potential security policy violations are performed as part of the process for accepting web responses into the protected local network. The product provides protection of an enterprise or local network environment in which additional policies and mechanisms are used to protect against the introduction of malicious code through means other than the web interface.

SurfinGate consists of three components: SurfinGate Server, SurfinGate Database, and SurfinConsole. Figure 1 illustrates the logical relationship between the SurfinGate Components. The SurfinGate Server interacts with the operating system to collect the web-based traffic. The SurfinGate Server, and SurfinGate Database work together to perform scanning of all the web-based traffic received from the network. The SurfinConsole runs on a separate platform and is used to manage the SurfinGate Server, and SurfinGate Database. There may be one or more SurfinGate Server components in the evaluated configuration.

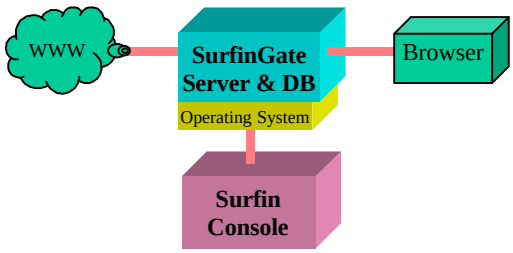

Figure 1: SurfinGate Logical Architecture

The SurfinGate Server is a proxy that runs on the Windows NT and the Sun Solaris operating systems. The proxy intercepts all inbound web-based traffic and performs a malicious mobile code analysis upon the web-based traffic. The

types of analysis supported include: content inspection of ActiveX, Java, Visual Basic Script and JavaScript, URL filtering, and file extension filtering. Any code that violates SurfinGate's security policy is denied access to the network.

The SurfinGate Database provides a repository for the security policy and security violations. The SurfinGate Database runs on the same platform as the SurfinGate Server. The SurfinGate Server stores all of its analysis results in the Database and receives all policy information from the database. The SurfinGate Server receives policy updates on a regular basis reflecting changes the administrator makes to the policy stored in the database. In the Windows NT configuration, the SurfinGate Database requires the presence of Microsoft Access on the Windows NT SurfinGate server platform and in the Solaris configuration, SurfinGate requires the Oracle Database Management System on the Solaris SurfinGate server platform.

The SurfinConsole is a central tool for managing the security policies, controlling multiple SurfinGate servers and generating audit reports. The SurfinConsole runs on a Windows NT 4.0 Service Pack 4 or above.

SurfinGate has the ability to filter web-based traffic searching for the following types of malicious mobile code:

- ActiveX,
- Java,
- Visual Basic Script, and
- JavaScript.

In addition to potential malicious mobile code, SurfinGate can filter web-based traffic based on the following criteria:

- File Extensions, and
- URLs.

With the collected web-based traffic, SurfinGate analyzes the data to see if it matches any identified risks in the security policy. The security policy is an administrator defined policy that specifies what types of data to filter and what action to take if data violates the security policy. The security violations are logged in a database (i.e., audit trail) for future analysis.

The SurfinGate system has the ability to audit and filter all web-based traffic. The SurfinGate system ensures the audit trail is protected and only administrators may view the audit data.

SurfinGate includes a number of management functions to control access to the system and to manage the data collection and analysis. The management functions include configuring the security policy that determines what information will be filtered and audited. Access to the management functions is restricted to the administrator. The administrator is the only user that directly accesses the SurfinGate product. The administrator is required to perform password authentication before accessing SurfinGate.

SurfinGate was evaluated in a network environment that restricted direct access to the SurfinGate components to the SurfinGate administrator. The evaluated configuration for SurfinGate is illustrated in Figure 2. SurfinGate resides in a network zone which is limited to web traffic and outbound e-mail. All HTTP requests from the browsers are proxied through SurfinGate server. The product release notes provide the administrator with specific instructions to ensure that the product is installed in an appropriate environment.

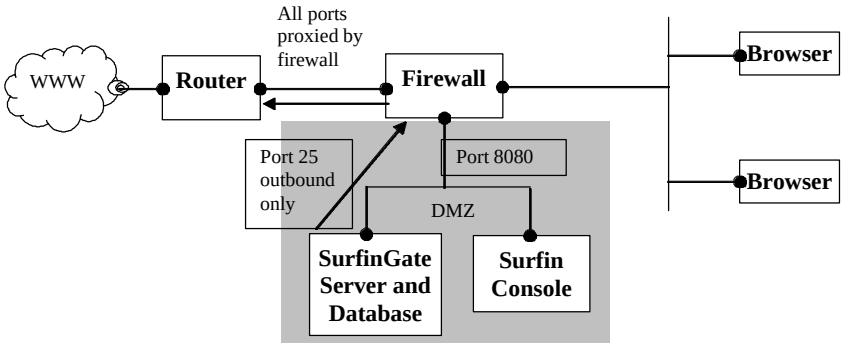

Figure 2: SurfinGate Evaluated Configuration

The validation team monitored the activities of the evaluation team, observed some team meetings with the developers, provided guidance on technical issues and evaluation processes, reviewed selected evaluation evidence, and reviewed the individual work units and successive versions of the ETR. The validation team found that the evaluation showed that the product satisfies all of the functional requirements and assurance requirements stated in the Security Target (ST). Therefore the validation team concludes that SAIC's findings are accurate, the conclusions justified, and the conformance results correct.

2 IDENTIFICATION

The CCEVS is a joint National Security Agency (NSA) and National Institute of Standards and Technology (NIST) effort to establish commercial facilities to perform trusted product evaluations. Under this program, security evaluations are conducted by commercial testing laboratories called Common Criteria Testing Laboratories (CCTL)s using the Common Evaluation Methodology (CEM) for Evaluation Assurance Level (EAL) 1 through EAL 4 in accordance with National Voluntary Laboratory Assessment Program (NVLAP) accreditation.

The NIAP Validation Body assigns Validators to monitor the CCTLs to ensure quality and consistency across evaluations. Developers of information technology products desiring a security evaluation contract with a CCTL and pay a fee for their product's evaluation. Upon successful completion of the evaluation, the product is added to NIAP's Validated Products List. Table 1 provides information needed to completely identify the product, including:

- the Target of Evaluation (TOE): the fully qualified identifier of the product as evaluated,
- the Security Target (ST), describing the security features, claims, and assurances of the product,
- the conformance result of the evaluation,
- the organizations and individuals participating in the evaluation.

Table 1: Evaluation Identifiers

Item	Identifier

Evaluation Scheme	United States NIAP Common Criteria Evaluation and Validation Scheme
Target of Evaluation	Finjan SurfinGate Version 5.6
Protection Profile	Not Applicable
Security Target	Security Target for Finjan SurfinGate 5.6, Version 1.7, October 25, 2001
Evaluation Technical Report	Evaluation Report for Finjan SurfinGate 5.6 Version 1.0, October 31, 2001
Conformance Result	Part 2 conformant, Part 3 conformant, and EAL 3
Version of CC	CC Version 2.1 [1], [2], [3] and all applicable National and International Interpretations effective on February 13, 2001
Version of CEM	CC Version 1.0 [7] and all applicable National and International Interpretations effective on February 13, 2001
Sponsor	Finjan Software Incorporated
Developer	Finjan Software Incorporated
Evaluators	Science Applications International Corporation Mr. Robert Williamson Mr. Mark Braga Government Participants None
Validators	Mr. Manilal Daya (The MITRE Corporation) Ms.Yi-Fang Koh (The MITRE Corporation) Dr. Jerome Myers (Aerospace Corporation)

3 SECURITY POLICY

SurfinGate performs filtering on web responses. When a response is received, SurfinGate collects it and sends it to its database for analysis. If a particular type of web-based traffic has been marked "disallow" in the security policy, the analysis engine will generate a violation and react as configured by the administrator. If a malicious mobile code element is discovered, the appropriate code scanner statically scans it. This scan function enables the analysis engine to determine if the code element is malicious.

When the SurfinGate Server discovers a violation, it can react in several ways depending upon the administrator defined security policy. The types of reactions are:

- Block – The SurfinGate Server can block the download of malicious mobile code, block web sites, and block data transfers;
- Alert – Send a visual alert to the user and/or email alert to the administrator; and
- Log – Simply log the violation and continue processing. In all cases a log record will be generated; however, in some instances, data transfer may or may not continue. SurfinGate enforces the following policies.

For the purposes of describing the security policies enforced by SurfinGate, these three categories of reactions are described separately below as a "Web Response Blocking", "Log", and "Alert" policies[1]. All three of these policies are configured and managed by the administrators through a common SurfinGate Console interface.

[1] This decomposition of the security policy is used in this document to briefly summarize the blocking, logging, and alerting features of the TOE security policy. The decomposition and terminology is not used in the Security Target or the vendor documentation. All three policies are combined in the Security Target as a "mobile code security policy" with the very similar expression "mobile code flow security policy" being used for the "web response blocking" aspect of the policy.

3.1 Client Identification

The information flow policy that is applied to clients is based upon an assumed identity of the client. Clients are identified by their network identities. SurfinGate server uses the destination host identified within the proxied “inbound” HTTP response packets as the identity of a client. The clients are categorized into groups of hosts for which specific "Web Response Blocking", "Log", and "Alert" rules may be associated. The categorization is tree structured by inclusion with the potential for explicit hosts being listed at the leaves. All hosts, regardless of whether they are explicitly listed within a grouping, are considered to be members of a “default” grouping that is the root of the tree.

3.2 Administrator Authentication

SurfinGate supports a number of policies and features that require appropriate management. SurfinGate provides for one role, the administrator. The security management functions are restricted to an administrator. Environmental restrictions ensure that only the administrator is permitted access to the platforms that host SurfinGate. SurfinGate provides an authentication mechanism to authenticate users to the TOE. Users must authenticate themselves to the SurfinGate Console locally. The login process prompts a user for a password. The user enters a password that is represented as “*” characters during authentication. That information is compared against a locally stored password. If the password matches the entry, the user is permitted access; otherwise, the user is denied access. All users that successfully authenticate to the SurfinConsole are administrators of the system.

3.3 Web Response Blocking

SurfinGate enforces an information flow control security policy, which filters web traffic based upon the attributes of the traffic and the identity of the destination for that traffic. The network traffic that is affected is "inbound" HTTP traffic, which the Security Target generically refers to as "response" data. Mobile code within response data may be "permitted", "marked for inspection", or "blocked". As described above in the "Client Identification" paragraph, the administrator defines a tree of groupings of clients. The access rights for some of the clients and groupings are explicitly specified. The access rights that are not explicitly defined for a client are inherited from the parent in the tree of client groupings.

The basic information flow security policy may be expressed by the following three rules:

1. If the mobile code type is explicitly marked as permitted, then the transfer of response data is permitted.
2. If the mobile code type is marked for inspection, the response data is permitted if no malicious mobile code is detected.
3. If the mobile code type is marked as blocked, then the response data is rejected.

All web-based response traffic is scanned for violations of the web response blocking security policy. The web response blocking policy can address the following types of mobile code:

- Java,
- JavaScript,
- Active X,
- VBScript

In addition, the policy can be configured to block transfers based upon:

- File Extensions, and
- URLs

The web response blocking security policy can permit or deny web-based traffic based on any of the characteristics identified above. The administrator selects which types of mobile code to filter and to what degree to perform filtering. Filters can be established in a general or specific manner. For example, the web-based traffic can be filtered based on a specific client or code type, or it can generally block a particular type of web-based traffic.

SurfinGate has the ability to block a specific script on a page while permitting the remainder of the page to be displayed. However, there is a caveat on that capability. In some cases, several objects upon which the filtering is based may be

contained within the same HTTP response. The TOE dynamically performs its analysis on the response data and strives to send permitted responses on to the destination client. If all of the mobile code elements within the response data are permitted, then the entire response will be sent to the client. However, if the TOE determines that a portion of a response should be rejected, then response data that preceded the rejected portion (in the response sequence) will still be delivered unmodified, but the blocked mobile code will be removed from the data and subsequent mobile code within the same response will also be similarly excised from the response. The removal of the subsequent mobile code is done in an attempt to gracefully close out the response by removing potential dependencies upon the rejected mobile code. The net effect is that some mobile code that might otherwise have been permitted will be blocked in that specific response.

3.4 Log Policy

SurfinGate creates audit records in accordance with an audit (or log) policy. The SurfinConsole is used to manage the audit log. When the analysis engine discovers a policy violation, an audit record is created and written into the audit log. The audit record contains the name of the requesting client, the type of mobile code, date, type of event, and success or failure.

The log policy is specified through the same interface as the response filtering policy. All web-based response traffic is scanned for violations of the audit policy. The log policy can address the following types of mobile code:

- Java,
- JavaScript,
- Active X,
- VBScript

In addition, web responses can be configured to log transfers based upon:

- File Extensions, and
- URLs

The log policy can result in an audit record being generated based on any of the characteristics identified above. The administrator selects which types of mobile code to log and to what degree to perform inspection. The administrator uses the same interface as for specifying the response filtering policy. Hence, the administrator may specify logging at the same granularity as blocking can be specified. Audit records are recorded for violations of the web response blocking security policy. However, the administrator may specify that logging will also occur when specific mobile code (or types of mobile code) is detected that is permitted by the response filtering policy.

The SurfinConsole is used to manage the audit log and to define the "Log Policy". In order to use the SurfinConsole, a user must be an administrator and must log in using a password. The SurfinConsole provides a graphical interface to manage, view, and analyze the audit logs.

3.5 Alert Policy

The TOE sends alerts based upon an alert policy. In the evaluated configuration, alerts take the form of electronic mail messages sent to a specific set of e-mail addresses. The set of e-mail addresses for alerts is common to all alerts and is configurable through SurfinConsole.

The alert policy is the similar to the "Log Policy" and is defined at the same time as the "Log Policy". The database that contains the client attributes for blocking and logging of mobile code also contains an attribute for sending an alert. If the attribute is set, then an alert will be sent to the specified e-mail addresses. Just as for audit, an alert may be sent even when mobile code is permitted.
Assumptions and Clarification of Scope

3.6 Usage Assumptions

The evaluation made the following assumptions concerning product usage

- There will be one or more competent individuals assigned to manage the TOE and the security of the information it contains
- The administrators are not careless, wilfully negligent, or hostile, and will follow and abide by the instructions provided by the TOE documentation.
- The product is configured according to the System Administrator's Guide and associated releases notes. In particular, the product is configured in a dedicated demilitarized zone in accordance with the environmental assumption described below.

3.7 Environmental Assumptions

SurfinGate was evaluated in a network environment that restricted direct access to the SurfinGate components to only the SurfinGate administrator. The evaluated configuration for SurfinGate is illustrated in Figure 2. SurfinGate resides in a network zone which is limited to web traffic and outbound e-mail. All HTTP requests from the browsers are proxied through SurfinGate server. The following is a slightly more complete summary of the evaluation assumptions concerning the environment:

- The processing resources of the TOE will be located within controlled access facilities, which will prevent unauthorized physical access.
- All of the SurfinGate related files and directories (including executables, run-time libraries, audit logs) are protected from unauthorized access by the underlying operating system and database control mechanism.
- SurfinGate will reside in a Demilitarized Zone (DMZ) to protect it from direct attacks.
- SurfinGate will be the only application running on its host server.
- A firewall will direct all web-based traffic through the SurfinGate product
- Users of the underlying operating system and database are identified and authenticated.
- The underlying system will provide mechanisms to isolate the TOE Security Functions (TSF) and assure that TSF components cannot be tampered with or bypassed. The TSF components are 1) the files used by SurfinGate to store its data and 2) the TOE processes managing the mobile code scanning.
- The mail server on the SurfinGate network will accept only outgoing mail from the SurfinGate product and will deliver mail properly.
- The host operating system for SurfinGate will only permit access by trusted users.
- The host operating system will provide a reliable timestamp.

3.8 Clarification of Scope

SurfinGate is not designed to resist direct, hostile attacks; therefore, it must be embedded in or protected by other products designed to address threats that correspond with the intended environment. Certain threats are outside the scope of the product's capabilities to counter, and the evaluation makes no claims of protection against them:

- The product provides protection from mobile code attacks for which the delivery of the mobile code would be forced to pass through the SurfinGate server. Protection from other means of delivery must be accomplished by other controls. In the evaluated configuration, this is accomplished by an environmental assumption that all mobile code entering the protected network was passed to the SurfinGate server for filtering. This requires that web browsers within the enclave are configured to proxy through SurfinGate and it has additional implications on the other protocols that may be permitted through the firewall. In particular, mail attachments need to be blocked or filtered by other means to effectively protect the hosts within the enclave. Similarly, if HTTPS traffic is permitted into the enclave, the TOE is not capable of inspecting the contents of the encrypted HTTP data packets and hence it cannot selectively block the delivery of HTTPS data based upon the contents of those packets. The environment must block the establishment of HTTPS sessions to avoid having undesirable mobile code wrapped within encrypted HTTPS response packets.
- The TOE implements its security policy based upon the identity of the destination host specified in the http headers of the packets that it receives. . The TOE relies upon the environment to correctly deliver packets that have been filtered.

4 ARCHITECTURAL INFORMATION

The product has three primary architectural components SurfinGate Server, SurfinGate Console, and the SurfinGate database. The product is designed such that the three components will operate correctly when installed on a single platform or when the SurfinGate Console is installed on a separate platform from the servers. The design allows for multiple instantiations of the server and database components. However, the database is expected to reside on the same base platform as one of the servers.

5 DOCUMENTATION

The product is distributed in shrink-wrapped packaging. The following product documentation is distributed along with a CD that contains the software:

- SFG_UG56 SurfinGate 5.6 for Windows NT/UNIX Solaris User Manual
- SFG_IG56 SurfinGate 5.61 for Windows NT/UNIX Solaris Installation Guide

In addition, the distribution media contains a "Readme" file that contains the release notes for the specific distribution. The "Release Notes for SurfinGate 5.61" are considered by the evaluation as part of the product documentation that should be read prior to installation of the product.

Additional unevaluated consumer documentation related to the product is available at http://www.finjan.com. This web site contains information that the vendor considers to be useful for informing its customers and potential customers about some explicit mobile code threats and the appropriate means of addressing those threats. Some of the information available at that site (at the time of the evaluation) was incorporated into the product evaluation as part of the product vulnerability analysis evidence. However, the content of the web site is dynamic and the information available at the web site is not considered to be part of the evaluation evidence.

6 IT PRODUCT TESTING

6.1 Developer Testing

The developer maintains a suite of tests for confirming that the product meets its advertised functional requirements. They tested the product in a networked environment using COTS browsers to drive many of the tests and COTS servers to provide the test data. The developer tests the product with a variety of platforms and configurations that includes some capabilities that were outside of the scope of the evaluated configuration. The evaluators reviewed the developers tests and test results to ensure that the developers testing and test results were appropriate for the evaluated configuration. The developer's Acceptance Test Plan (ATP) contains a list of all security-relevant tests. This ATP includes test cases for all external interfaces and all security functions.

The external interfaces identified include the following:

- Console interface
- Mail Server
- Security database
- Administrator interfaces
- Operating systems
- Logs and error interfaces.

A mapping was provided between the interfaces and the ATP, and at least one test case was mapped to every external interface. Many of the interfaces are exercised in multiple tests. For instance, once the administrator successfully authenticates to the console, every test is performed by the administrator following the “User Guide” to establish specific permissions and attempting to run specific types of mobile code through SurfinGate from pre-saved web pages. Hence, almost every test exercises the console interface. Similarly every test that makes a policy decision exercises the security

database interface. Also, like the security database tests and the console tests, the operating system interface is exercised through each test case.

The evaluation team reviewed all of the security functions and the mapping between security functions and tests. All security functions were appropriately tested.

6.2 Evaluator Testing

The evaluators performed several testing activities associated with this product. The evaluators performed some testing with the assistance of a vendor representative and the evaluators independently repeated a subset of those tests.

The test environment required the use of COTS web browsers to exercise the security policy and the use of the administrative interface provided by SurfinConsole to configure the security policy and to confirm the results from some of the tests. Both of those interfaces are GUIs. Hence, the tests and the verification of the test results typically required the manual observation of a result presented by the GUIs. The testers were provided detailed instructions for observing and recording the test results. However, the actual execution of the test suites was cumbersome and human errors were easily introduced into the tests if they were exercised too quickly. Hence, the testing was carefully observed and some of the tests had to be repeated due to data entry errors or steps being skipped in the procedures.

The evaluators worked with a member of the development team to exercise an evaluation team selected suite of tests from the ATP. Every security function identified in the Security Target was exercised in that subset of the vendor testing. The evaluators confirmed that the tests produced the expected results. Since the initial execution of the developer tests included the participation of a member of the development team, the evaluators also performed a separate testing exercise in which they ran approximately 25% of the developer's test suite as well as all supplemental evaluation team tests. The subset that was tested without developer participation consisted of a broad sample that tested each security function in the Security Target.

The evaluators performed some independent product testing of their own design. Through analysis of the vendor test suite, the evaluators determined that the vendor testing was thorough enough to confirm that the TOE provided the functionality claimed in the ST. Hence, they did not develop a large number of independent tests. However, there was some concern that all of the test data that was used in the developer's tests to exercise the mobile code detection and filtering capabilities of the product was controlled by the developers. In particular, there were no vendor tests on "live" data. Although this is appropriate for ensuring that test data contained known forms of mobile code that properly tested the TOE functionality, it also left open the possibility that the testing was not robust and the product would not detect similar mobile code during "live" web browsing. The evaluators were concerned that, even though they had reviewed the vendor tests to ensure that the tests were appropriate, there might be a bias in the vendor test data that the evaluators had missed. Hence, the evaluators identified some additional web content on the Internet that was thought to contain the types of mobile code that SurfinGate claimed to detect. The evaluators analyzed the content of those web sites to confirm that it did indeed contain the desired types of code. They tested the SurfinGate filtering claims on that data, and they confirmed that the correct results were obtained.

All tests gave expected (correct) results. The testing found that the product was implemented as described in the functional specification and did not uncover any undocumented interfaces or other security vulnerabilities.

7 EVALUATED CONFIGURATION

The TOE was evaluated in two configurations:

- SurfinGate Server and SurfinGate Database are installed on the same hardware platform with the Windows NT operating system and an Access Database. SurfinConsole is installed on a hardware platform with the Windows NT operating system. The platform for SurfinConsole can be the same as the platform that hosts SurfinGate Server and SurfinGate Database
- SurfinGate Server and SurfinGate Database are installed on the same or different hardware platforms with the Sun Solaris operating system on both platforms and Oracle installed as the DBMS for the SurfinGate Database. SurfinConsole is installed on a separate hardware platform running the Windows NT operating system.

The environmental and usage assumptions described in this report are applicable to both evaluated configurations.

8 RESULTS OF THE EVALUATION[2]

The evaluation was conducted based upon CC version 2.1 [1], [2], [3] and CEM version 1.0 [7] and all applicable National and International Interpretations in effect on February 13, 2001. The evaluation determined the product to be **Part 2 conformant, Part 3 conformant,** and to meet the requirements of **EAL 3**. The details of the evaluation are recorded in "Evaluation Technical Report for Finjan SurfinGate, Version 1.0, October 31, 2001" which is controlled by SAIC.

9 EVALUATOR COMMENTS

There are no Evaluator Comments.

10 ANNEXES

There are no annexes to this report.

11 SECURITY TARGET

The Security Target, "SurfinGate 5.6 Security Target, ST Version 1.7 October 25, 2001", is included here by reference.

[2] The terminology in this section is defined in CC Interpretation 008, specifying new language for CC Part 1, section/Clause 5.4.

12 GLOSSARY

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
EAL	Evaluation Assurance Level
EDR	Evaluation Discovery Report
ETR	Evaluation Technical Report
HTTP	Hyper Text Transfer Protocol
IP	Internet Protocol
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
SAIC	Science Applications International Corporation
SAR	Security Assurance Requirement
SFR	Security Functional Requirements
SOF	Strength of Function
ST	Security Target
TOE	Target of Evaluation
TSF	TOE Security Functions
TSFI	TSF Interface

13 BIBLIOGRAPHY

[1] Common Criteria for Information Technology Security Evaluation – Part 1: Introduction and general model, dated August 1999, Version 2.1.

[2] Common Criteria for Information Technology Security Evaluation – Part 2: Security functional requirements, dated August 1999, Version 2.1.

[3] Common Criteria for Information Technology Security Evaluation – Part 2: Annexes, dated August 1999, Version 2.1.

[4] Common Criteria for Information Technology Security Evaluation – Part 3: Security assurance requirements, dated August 1999, Version 2.1.

[5] Common Evaluation Methodology for Information Technology Security – Part 1: Introduction and general model, dated 1 November 1998, version 0.6.

[6] Evaluation Technical Report for Finjan SurfinGate, Version 1.0, October 31, 2001

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

[8] Evaluation Technical Report for Finjan SurfinGate, Version 1.0, October 31, 2001

[9] SurfinGate 5.6 Acceptance Test Plan, dated July 4, 2001

[10] SurfinGate 5.6 Security Target, ST Version 1.7 October 25