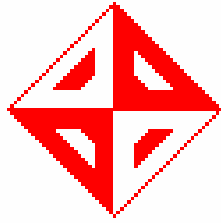


# Software Test Report

May 28<sup>th</sup>, 2006

SmartGuardian

**SmartGuardian™ Software**



**Computer Engineering Department  
Middle East Technical University**

by

SmartSoft Network Solutions, Inc.

**Tayfun Şen - Özden Meren - Sedat Zelyüt - Hakan Özadam - Akın Öztürk**



## **TABLE OF CONTENTS**

<b>1. Unit Testing.....</b>	<b>3</b>
<b>1.1 Controller Testing.....</b>	<b>3</b>
<b>1.2 Wall Testing.....</b>	<b>7</b>
<b>2. Integration Testing .....</b>	<b>11</b>
<b>3. Performance Testing.....</b>	<b>11</b>
<b>4. Volume Testing.....</b>	<b>12</b>

# 1. Unit Testing

## 1.1 Controller Testing

Testing of Controller Module is composed of four subtasks. First subtask checks for authentication and time expiration characters of Controller. Second subtask considers **User, User Group & Admin** options; adding deleting and modification consideration. Third subtask interest in Wall module's options. This subtask checks domain classes, domains in domain classes, policies & rules in policies; modifications in the settings and its inconsistency with the database (DB). Fourth subtask examines Reporter submodule of Controller module. Testing has been done through PHP pages of Controller and MySQL database whether the changes affects DB or not.

<b>Test Case ID</b>	Task-1			
<b>Start Condition</b>	Browser open Controller page for display			
<b>Description</b>	Authentication & Expiration			
<b>Due Date</b>	11.05.2006			
<b>Step</b>	<b>Test Data Used</b>	<b>Expected Result</b>	<b>Pass</b>	<b>Fail</b>
1	Authenticate to system with a user name & password which are exist in DB and correct	Authenticate to Controller Remote System successfully	***	
2	Authenticate to system with correct username & wrong password & vice versa	Failure in authentication	***	
3	Authenticate to system with wrong username & password	Failure in authentication	***	
4	Wait for 2 minute without any process in Idle state	Logout and display Log in page for Controller due to expiration	***	
5	Click Logout link	Logout and display Log in page for Controller	***	
<b>End of Test Case</b>				

<b>Test Case ID</b>	Task-2
<b>Start Condition</b>	Clicking <u>Settings</u> link
<b>Description</b>	Testing User Groups, All Users, Admin fields for modification consideration
<b>Tester</b>	
<b>Due Date</b>	13.05.2006

<b>Step</b>	<b>Test Data Used</b>	<b>Expected Result</b>	<b>Pass</b>	<b>Fail</b>
1	Clicking <u>User Group</u> link & inserting a new user which is client to SG	Insert into DB correctly with appropriate attributes and refresh page to show	***	
2	Modification on a user that exist in DB (add, delete, update)	Modification exactly affects the DB	***	
3	Modification on a user group that exist in DB (add, delete, update)	Modification exactly affects the DB	***	
4	Changing policy for a specific user group by update	Change is effect the attribute of appropriate table	***	
5	Clicking <u>All Users</u> link	Displaying all users in same frame and modification options	***	
6	Clicking <u>Admin</u> link & apply modification	Displaying all administrators in SG & any modification effects system refresh mechanism work	***	
7	Modification application of an admin that do not have privileges to do modification in ADMIN table	Ought to be check for privileges		***
<b>End of Test Case</b>				

<b>Test Case ID</b>	Task-3
<b>Start Condition</b>	Clicking <u>DomainClasses</u> & <u>Policy links</u>
<b>Description</b>	Testing Domain Classes & Policy fields for view and modification consideration
<b>Tester</b>	
<b>Due Date</b>	12. 05.2006

Step	Test Data Used	Expected Result	Pass	Fail
1	Clicking <u>DomainClasses</u>	Displaying all domain classes and have a link to each Domain class page	***	
2	Clicking <u>Policy</u> option and apply modification to policies	Displaying all policies in a table and process modification to Policy table in database and refresh page	***	
3	Adding & Deleting a Rule	Apply this in DB without causing inconsistency in DB	***	
4	Pressing a <u>RuleEdit</u> link and apply modification to Rule which is unique for that policy	Displaying that Rule's page in a frame and execute modification to change DB	***	
5	Make changes in Rule attributes and press update button	New attributes set in That Rules attribute and refresh page with new attributes	***	
6	Add & delete Time Range	Apply this in DB without causing inconsistency in DB	***	
7	Add & delete Port Range	Apply this in DB without causing inconsistency in DB	***	
8	Add & delete Application	Apply this in DB without causing inconsistency in DB	***	
9	Add existing File Rule	Just effect Rule_FileRule Table with a new row	***	

10	Add existing Domain Classes	Just effect Rule_DomainClasses Table with a new row	***	
11	Add New File Rule	Apply this in DB without causing inconsistency in DB	***	
12	Edit File Rule	Displaying File Rule page with extensions & keywords	***	
13	Add & delete Extension & Keyword	Apply this in DB without causing inconsistency in DB	***	
14	Add New Domain Class	Expected to add Domain Class as a new row	***	
15	Add/delete Domain to/from Domain class	Expected to do it for Domain & Have_Domain tables	***	
16	Click Go back from File Rule & Domain Class	Due to not coded yet expected to fail		***
17	Clicking Go Back from PortRange, TimeRange, Application, Protocol	Expected to turn that Rules frame after refresh all	***	
<b>End of Test Case</b>				

<b>Test Case ID</b>	Task-4			
<b>Start Condition</b>	Clicking <u>Reporter Section &amp; Statistics</u> subsection			
<b>Description</b>	Displaying User behavior in a way to displaying all requested Url's time and size values			
<b>Tester</b>				
<b>Due Date</b>	13.05.2006			
<b>Step</b>	<b>Test Data Used</b>	<b>Expected Result</b>	<b>Pass</b>	<b>Fail</b>
1	Click Display All Statistics	Displaying all action for all		

	& click a URL link in there	user and actually for small time interval & open clicked URL link in browser	***	
2	Click User Statistics	Display in a column all active users And have a link to all users statistics page	***	
3	Click a User	Display all URLs, time, and size of that users activities on INTERNET	***	
4	Click Search Statistics	Displaying search engine page with search by IP, Username & URL options	***	
5	Just choose one of three and press search options	Displaying statistic according to search option if exist	***	
6	Combine or three of them and press search option	Display intersect statistics of typed search options	***	
7	Click delete Statistics	Displaying delete page with delete by IP, Username & URL options	***	
8	Just choose one of three and press delete options	Deleting rows that mach delete options for delete options	***	
9	Combine or three of them and press delete option	Deleting intersected rows that mach delete options for delete options	***	
<b>End of Test Case</b>				

## 1.2 Wall Testing

Testing of Wall Module is composed of three subtasks. First subtask checks for authentication and time expiration characters of Wall. Second subtask considers **User**, **User Group** options; adding deleting and modification consideration in database can be adapted to Wall or not & Rules in policy which uniquely map user groups behave correctly. Third subtask is about each attribute of rule and any related attributes such as time range, port range, domain classes or file rules. In this subtask, these attributes are checked to be used by Wall or not.

<b>Test Case ID</b>	Task-5
<b>Start Condition</b>	Browser open to have request through Internet
<b>Description</b>	Authentication & Expiration
<b>Due Date</b>	11.05.2006

<b>Step</b>	<b>Test Data Used</b>	<b>Expected Result</b>	<b>Pass</b>	<b>Fail</b>
1	Authenticate to system with a user name & password which are exist in DB and correct	Login successful page will appear on the browser and any request from that IP will be put on process	***	
2	Authenticate to system with correct username & wrong password & vice versa	Login unsuccessful page will appear and if any request is done again authentication page will be displayed in a loop	***	
3	Authenticate to system with wrong username & password	Login unsuccessful page will appear and if any request is done again authentication page will be displayed in a loop	***	
4	Wait for 3 minutes (might be changed) without any process in Idle state	Due to a definite time without doing nothing time expiration will succeeded & the client will be logged out	***	
5	Type browser address row <b>"log.me.out"</b>	The user automatically be logged out and any request after will be denied	***	
<b>End of Test Case</b>				



Test Case ID	Task-6
--------------	--------

<b>Test Case ID</b>	Task-6
<b>Start Condition</b>	Changes in User Group and User and pressing <i>flushdatabase</i> button for authenticator
	Authenticator Submodule Testing
<b>Due Date</b>	11.05.2006

Step	Test Data Used	Expected Result	Pass	Fail
1	Adding a new user in a database and flush it	After system read whole database if it flushed then any authentication will be accepted	***	
2	Deleting an existing user though controller and flush it	After system read whole database if it flushed then any authentication will be rejected	***	
3	Changing User Group of a User or Policy of a User group in which that user exists	Behavior of system against that user will be changed according to new policy rules.	***	
<b>End of Test Case</b>				

<b>Test Case ID</b>	Task-7			
<b>Start Condition</b>	Changes in Domain Classes Policies & Rules and pressing <i>flushdatabase</i> button for Verifier			
	Verifier Submodule Testing			
<b>Due Date</b>	11.05.2006			
<b>Step</b>	<b>Test Data Used</b>	<b>Expected Result</b>	<b>Pass</b>	<b>Fail</b>
1	Checking Precedence of Rule in a policy by updating two Rule both and observe which is considered before	Higher precedence attribute will be considered first but if it does not match than the other rule will be applied	***	
2	Updating a rules attribute for example make it allow it deny	Then if it deny news site than after update and flush the behavior will be changed according	***	
3	Update & add Time Range of a high precedence Rule	If rule deny than with new interval it will apply if match	***	
4	Port Range & Application & Protocol changes	Due to not coded it will not work		***
5	Deleting a Domain Class from higher precedence rule	It will be observed that if rule say deny then domains in deleted domain class will be reached by client and vice versa.	***	
6	Adding a Domain Class to higher precedence rule	It will be observed that if rule say deny then domains in added domain class will be denied also by system and vice versa.	***	
7	Add a extension jpg to a file rule in a rule in system	If rule deny then the system deny file with jpg extension file	***	
8	Add a keyword to a file rule like "fun"	Then if rule deny; afterwards it will prevent downloading a file that have "fun" as substring	***	
9	Deleting a file rule and flushing database	Each extension and key words for file -if rule deny- will be allowed vice versa	***	

10	Changing logged attribute of a Rule then flush database	After that change the action of client will be wrote into database	***	
<b>End of Test Case</b>				

## 2. Integration Testing

In first part, communication of Controller with Wall module of SmartGuardian software is done by database of system. Any changes in the database of the system through Controller are detected by the Wall module. This is accomplished by first making changes in Authenticator tables (User, User Group, Admin etc.) or Verifier tables (Domain Class, Policy, Rule, File Rule etc.) and then changing dirty table flag attribute. The Wall module then checks this table every minute and recognizes any changes; it then reads the whole table again. In average, in 30 seconds any changes will adapted to the system.

In the second part clients and behavior of Wall to each client tested. Each client is from different user group and therefore different policies are executed for each demand of web pages from system, this is checked to be as expected or not. Results of this test is also found to be satisfactory.

Third part of testing involves Crawler's integration with the system. Due to it not being completed yet; testing of this module is being done in functionally. Integration will be done later.

## 3. Performance Testing

This testing is conducted to evaluate the compliance of the SmartGuardian system and/or its components with the specified performance requirements. Often this is performed using an automated test tool to simulate large number of users. Also known as "Load Testing", this testing has been done in three phases.

In the first phase, single user demand with large amount of requests from system is tested. Performance of the system is quite satisfactory and speed of response to client browser is nearly the as same as without our system in between. Second phase is tested with much number of clients, nearly 30 (due to small amount of tester) & each with small number of requests from system. This test result was also satisfactory. Last step was with large amount of user & each with large amount requests. It is also satisfactory for performance of system. Since all system have some limit number to serve and break out. It might happen due to hardware or software problems. Such tests will be applied in near future.

Beside the performance, durability of the system is also tested in testing process. Three hours without any interruption or failure of system, it has continued to serve clients. Such test shows that the system durable.

## **4. Volume Testing**

Results of testing confirmed that any values that may become large over time (such as accumulated counts, logs, and data files), can be accommodated by the program and did not cause the program to stop working or degrade its operation in any manner. The system accumulates a lot of data and keeps these in the database, so over time the volume of the data can become very large. In the tests we have seen that this does not slow down the SmartGuardian system, but it may result in problems because of hardware limitations, such as disk size.