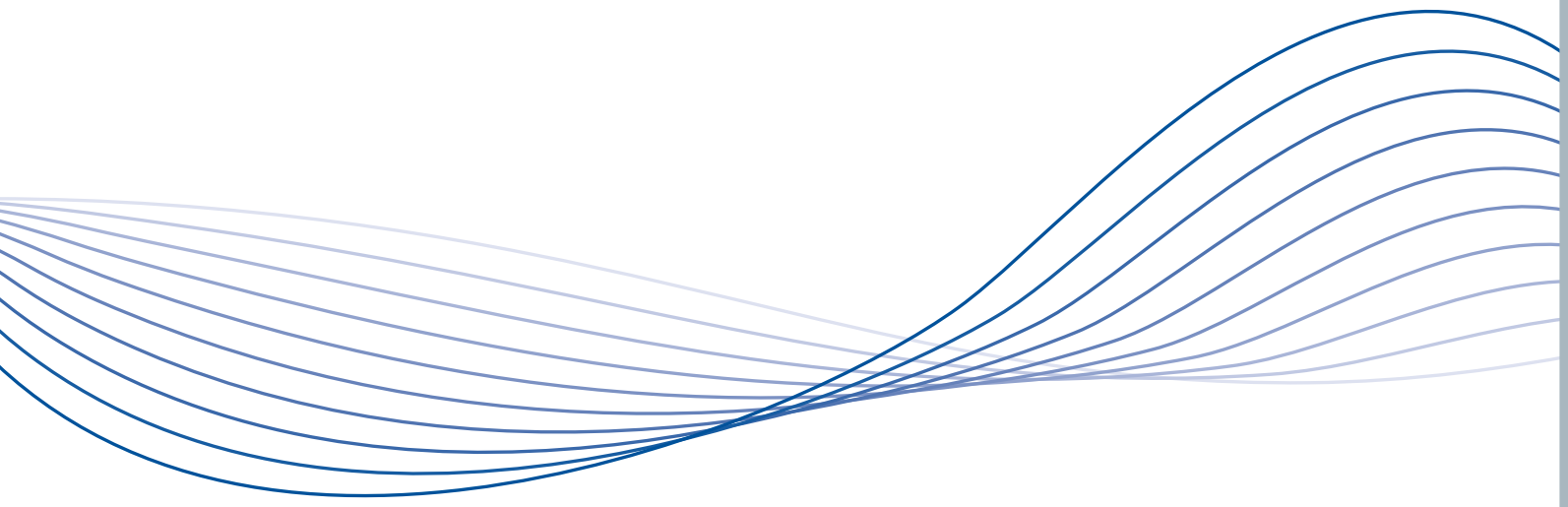
AppSense®

User Guide



AppSense®
Security Analyzer

Product Version: AppSense Security Analyzer
Publication: 1

Document Version: APSA61-01-030206-1

Contents

About the Guide

This guide provides you with instructions for using Security Analyzer on a typical system. The guide explains the various components of Security Analyzer and elaborates on the tests that Security Analyzer performs.

AppSense Security Analyzer is an application that performs a variety of tests on your system and highlights areas of vulnerability. In its most simple use, you can run all the available tests and get a quick overview of what vulnerabilities were found on your system.

Requirements

AppSense Security Analyzer has the following system requirements:

- Windows 2000 or greater
- .NET Framework 1.1
- Internet Explorer 5.5 (or greater)
- Internet connection (for internet tests)

Using the Analyzer

AppSense Security Analyzer is available as a free download from the AppSense website.

http://www.appsense.com/SecurityAnalyzer

➲ To use Security Analyzer in a typical scenario:

1. Download the program and place it in a location on the local machine that is accessible to all users. The following path is recommended to provide easy access:

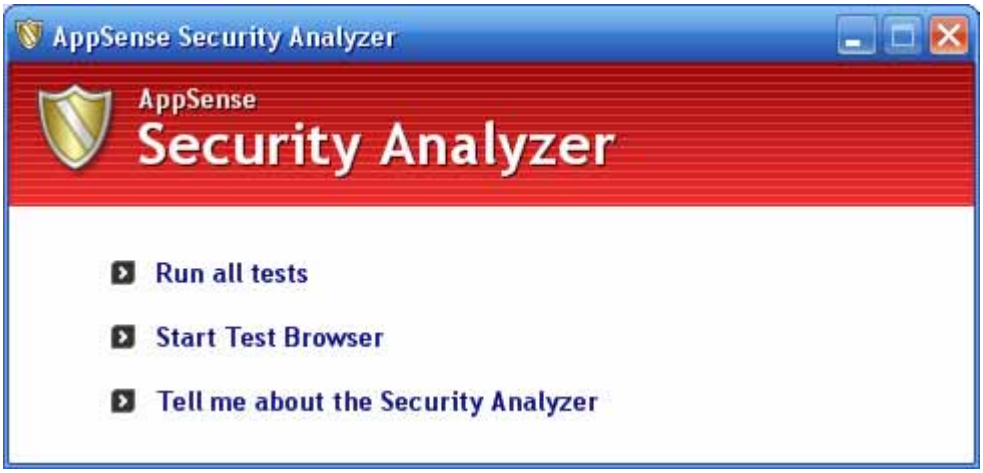
   %SYSTEMDRIVE%\Documents and Settings\All Users\Desktop

Note Security Analyzer is intended to be run in the context of a typical user account on the local machine, to highlight the fact that an ordinary user has permission to perform actions that could be potentially damaging to the system. These types of actions may be performed accidentally or maliciously by the user, malware is often designed to execute its actions in a covert manner.

Warning Before running Security Analyzer close all running applications. When running the Security Analyzer tests various windows will appear, this is a normal part of the test process.

2. Once downloaded, double click the executable to launch Security Analyzer.

Main Screen



The main screen provides the following options:

- **Start Tests**

  Select to automatically run all of the security tests. Click Stop Tests at anytime to abort. On completion of the tests the Test Browser displays.

- **Start Test Browser**

Select to launch the Test Browser. This allows you to obtain information on individual tests and execute each test in sequence.

- **Tell me about the Security Analyzer**

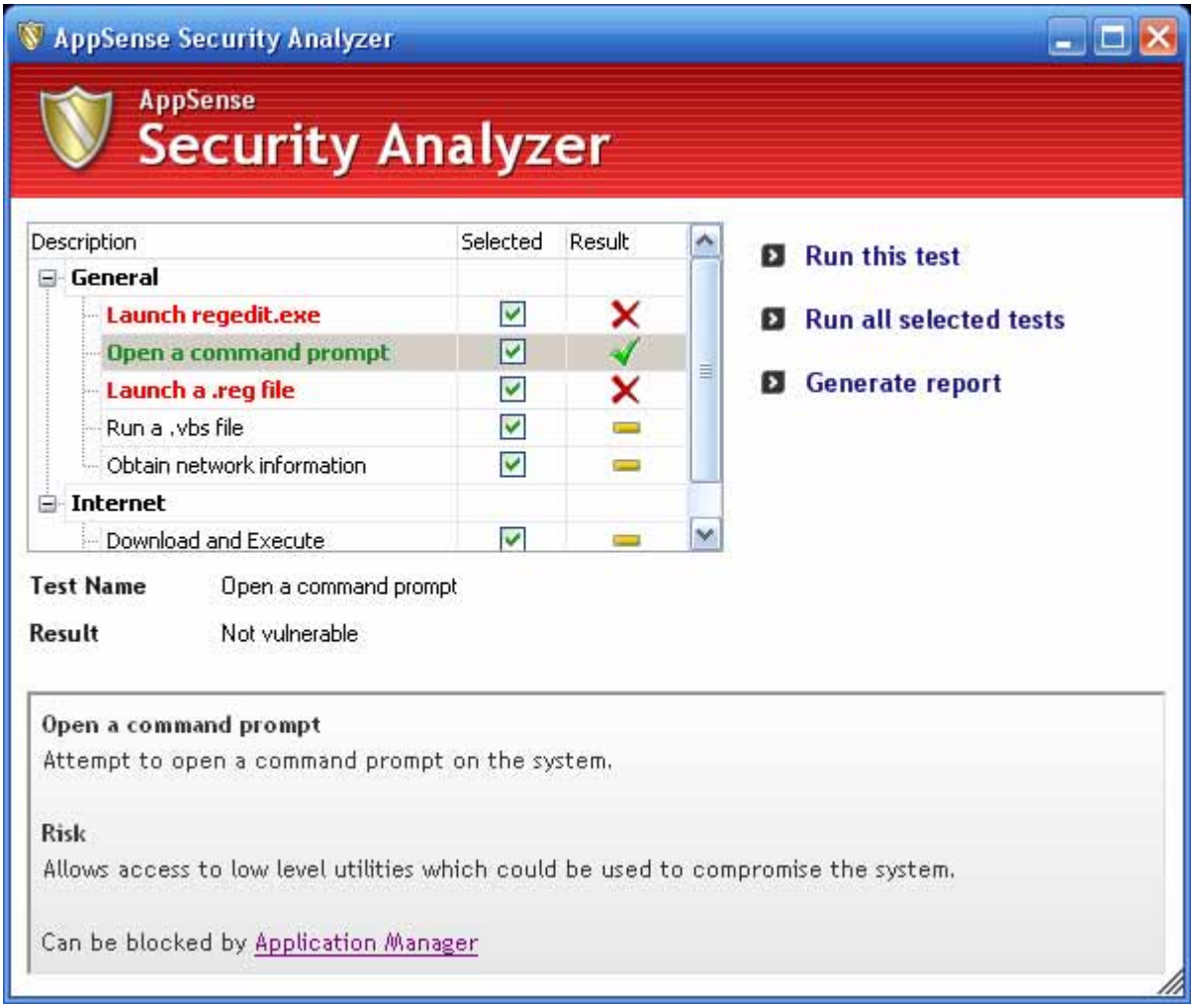
  Select to launch the Security Analyzer Help.

  Note This requires Acrobat® Reader®.

To close Security Analyzer use **ALT+F4**, or click **Close**.

The Test Browser

The test browser allows you to select and gather more information about individual tests. You can run one or more tests. This screen shows the current outcome for each test:



The test browser screen is divided into the following 3 areas:

Top Left – Test Navigation.

All available tests are shown in the tree view. Select a test to display information about the test. Double-click a test to run the test now. If the test has not run yet, the name is shown in black. Once a test has run once, the color changes as follows:

Red – Failed. The test has found a vulnerability on the system.

Green – Passed. The test has not found a vulnerability.

Orange – Skipped. The test is unable to complete.

When a test is shown as **Skipped**, you can determine why it was skipped by selecting the test and examining the result. Using the Selected checkbox column you can create a subset of tests to run when clicking the **Run all selected tests** action.

Top Right – Actions

This section contains three actions you can perform from the test browser:

Run this test – Run the currently selected test.

Run all selected tests – Run each test that has its **Selected** option ticked in the tree view.

Generate report – Launch a new window containing a report of the test results. This can be run at any time.

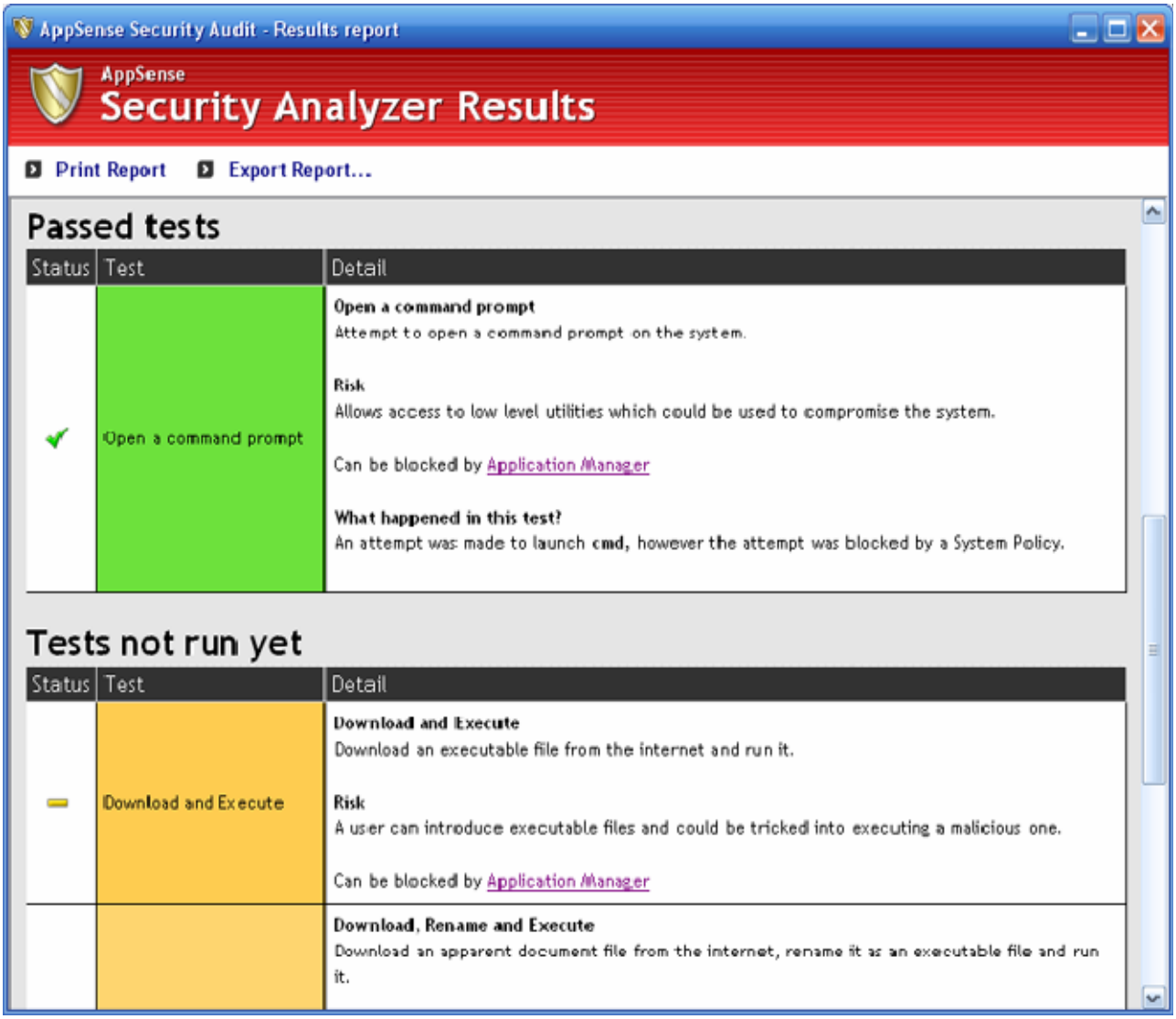
Note You can display multiple report windows simultaneously for making comparisons.

Bottom – Test Information

The area at the bottom of the test browser displays information about the test currently selected in the tree view. You can see the test name, current result and a more detailed description of the test. When a test has run and failed, the word **Vulnerable** displays in red as the result.

Reports

From the test browser, you can generate a report that provides a concise view of the current test results. Click **Generate report** to display a report similar to this:



The tests are shown in a specific order in the report:

- The tests that failed (exposed vulnerabilities on the system).
- The tests that passed (no vulnerability).
- The tests that were skipped.
- The tests that have not run yet.

Each item in the report has a '**What happened in this test?**' section that expands on the test outcome and in the case of skipped tests briefly explains why the test was skipped.

The Tests

Launch regedit.exe

This test shows that when running Security Analyzer you can open the standard Windows system registry editor. The ability to run the registry editor implies you can:

- Explore the registry and gain information about the system and its setup.
- Potentially add or overwrite existing registry data which is not secured via permissions.

How the Test is Performed

- It simulates launching regedit.exe, much in the same way as **Start >Run**, type **regedit** and Enter.

How the Result is Determined

- Passed – regedit failed to execute.
- Failed – regedit executed successfully.

Download and Execute

This test shows that when running Security Analyzer you are able to download an executable file (.exe) from the Internet. Once that file is downloaded to a directory where you have permission to write, the test demonstrates that the executable file can be launched.

How the Test is Performed

1. The test checks for the existence of a remote web file:

   http://www.appsense.com/files/demos/security_analyzer/Sample1004.exe

   Note This file is a harmless executable hosted on our website.

2. Action is taken as follows:
   - If the file doesn't exist the test is aborted and the result set to skipped. Additional detail is provided explaining that the file could not be downloaded.
   - If the file exists, the file is downloaded over HTTP protocol to the path pointed to by the **temp** environment variable. For example:

     C:\Documents and Settings\user\Local Settings\Temp

     If the download fails, the test result is set to skipped and additional detail is provided explaining that the file could not be downloaded.
3. Assuming the file is downloaded, an execute attempt is made and the file is deleted.

How the Result is Determined

- Passed –Downloaded file failed to execute.
- Failed –Downloaded file executed successfully.

- Skipped – The file could not be downloaded.

Download, Rename and Execute

This test is similar to the Download and Execute test, it shows that you can download executable content from the internet. In this test, the downloaded file is hosted remotely as a document (.doc). This test shows that local settings or a firewall/proxy may be tricked into allowing the downloading of executable content.

How the Test is Performed

1. The test checks for the existence of a remote web file:

   http://www.appsense.com/files/demos/security_analyzer/Sample1006.doc

   Note This file is a harmless executable hosted on our website, renamed as a document.

2. Action is taken as follows:
   - If the file doesn't exist the test is aborted and the result set to skipped. Additional detail is provided explaining that the file could not be downloaded.
   - If the file exists, the file is downloaded over HTTP protocol to the path pointed to by the **temp** environment variable. For example:

     C:\Documents and Settings\user\Local Settings\Temp

     If the download fails, the test result is set to skipped and additional detail is provided explaining that the file could not be downloaded.
3. Assuming the file is downloaded, it is renamed from Sample1006.doc to Sample1006.exe.
4. An attempt is then made to execute Sample1006.exe.

How the Result is Determined

- Passed –Downloaded file failed to execute.
- Failed –Downloaded file executed successfully.
- Skipped – The file could not be downloaded.

Open a Command Prompt

This test shows when running Security Analyzer you can launch the Windows command prompt.

How the Test is Performed

- It simulates launching **cmd.exe**, much in the same way as **Start >Run**, type in **cmd** and Enter.

How the Result is Determined

- Passed – Command prompt failed to launch.
- Failed – Command prompt launched successfully.

Launch a .reg File

This test shows when running Security Analyzer, you can create and launch files that populate or manipulate registry data. This test uploads data to the **Winlogon** key (for the current user).

Note This key is often used to place entries that launch malicious software when a user starts up their computer.

How the Test is Performed

1. A new file is created in the directory pointed to by the **temp** environment variable. For example:

   C:\Documents and Settings\user\Local Settings\Temp

   The file created is named **securityAuditTestREG.reg** and contains the following:

```
Windows Registry Editor Version 5.00

[HKEY_CURRENT_USER\\Software\\Microsoft\\Windows NT\\CurrentVersion\\Winlogon\\Test]

"Test"=dword:00000000
```

   This .reg file creates a registry value named **Test** under the **Winlogon** key.
2. The created .reg file is executed as an argument to **regedit**.
3. The test removes the new registry entries if it succeeded.

How the Result is Determined

- Passed – New data was not entered into the system registry.
- Failed – New data was entered into the system registry.

Run a .vbs File

This test shows when running Security Analyzer, you can knowingly or unknowingly run scripts. Along with their many legitimate uses, scripts are commonly used by viruses and worms. In many cases, scripts that form part of a virus can launch without you being aware they have launched.

How the Test is Performed

1. A file is created in the directory pointed to by the **temp** environment variable. For example:

   C:\Documents and Settings\user\Local Settings\Temp

   The file created is named **temp.vbs** and contains the following:

```
Set FSO = CreateObject("Scripting.FileSystemObject")

Set TextStream = FSO.CreateTextFile(<usertemppath> & “\testScriptOutput.txt")

TextStream.WriteLine("output")

TextStream.Close
```

This script attempts to create a file in the directory pointed at by the temp environment variable.

2. **wscript.exe** is launched with the generated script file path as an argument. For example:

```
wscript.exe “c:\documents and settings\user\Local Settings\temp\temp.vbs”
```

3. On completion of the test the created files, the script and the file the script may create if it can execute are deleted.

How the Result is Determined

- Passed – If the file that the script intended to create after the script was executed (testScriptOutput.txt) doesn’t exist, the test is deemed to have passed as no file was created.
- Failed – If the file that the script intended to create does exist after the script is executed the test is deemed to have failed, because the script was able to carry out its function.

Obtain Network Information

This test shows when running Security Analyzer, you can use tools that already exist on the system to collection information regarding network connectivity. This information could be used by a malicious user to identify security issues on a corporate network. This test runs the tools via the command prompt, and will still function even if a system policy is in place that prevents command prompt access.

How the Test is Performed

1. This test runs 3 tools that already exist on a Windows system; **hostname**, **ipconfig** and **route**.
2. The test runs each tool and collects information from the tools into a temporary file.

How the Result is Determined

- Passed – No data was collected.
- Failed – At least one or more pieces of network information were collected.