Welcome to the Attack Surface Analyzer wiki!

Attack Surface Analyzer 2.3 is now available

The project main branch is now moved up to 2.4 for ongoing improvements.

Why Attack Surface Analyzer

Attack Surface Analyzer (ASA) is a Microsoft-developed Security tool that analyzes the attack surface of a Windows, Linux or MacOS system and reports on system changes that may have potential security implications that are introduced by the installation of software or by system misconfiguration.

Attack Surface Analyzer 1.0 from Microsoft was released in 2012 and is no longer available. Attack Surface Analyzer 1.0 has been valuable to software developers and IT security personnel for years in helping detect key system changes that may occur from software installation.

Attack Surface Analyzer 2 is a rewrite from the ground up on .NET Core and is an Open Source project managed by Microsoft.

Scenarios

1. Attack Surface Analyzer can help identify potential security risks exposed through changes to services, user accounts, files, network ports, certificate stores, and the system registry. It also includes some support for “live” monitoring of certain system changes (i.e. file system and registry).
2. Another key use for the tool is in ensuring your software development process and products are following best practices for least privilege and reducing the attack surface for your customers by providing evidence, to your security and release teams, that your code does only what it claims. Maintaining customer trust is one reason why it is recommended from the Microsoft SDL Practices.

Typical users of ASA:

- DevOps Engineers - View changes to the system attack surface introduced when your software is installed.
- IT Security Auditors - Evaluate risk presented when third-party software is installed.

How to Run Attack Surface Analyzer

Overview

Attack Surface Analyzer 2 comes with both a command line (CLI) or an browser based (GUI) option, making it easy to use as part of a testing or release script or for stand alone use. When using it, you create “snapshots” before and after you install the target software under consideration for analysis. A clean initial system with minimal additional software is ideal, but not required. Snapshots are stored in a local SQLite database and used to generate reports of system changes.

You can also scan for changes after the software is used or while it is running to potentially capture additional changes made to the system.

Note: Attack Surface Analyzer requires administrator privileges to accurately gather system data.

The basic steps for running Attack Surface Analyzer are:

1. Take a baseline scan on a clean machine.
2. Install and run your product or application. Optionally make these two separate scans to distinguish between install vs run changes that are made.
3. Take a product scan.
4. Run data analysis.

The assumption is that both data collection and data analysis will be run on the same machine and that the same elements are collected in the baseline and subsequent scans.

Installation

Nuget Packages

Attack Surface Analyzer is distributed on Nuget in CLI and Library form.

If you have .NET Core installed, you can install Attack Surface Analyzer with `dotnet tool install -g Microsoft.CST.AttackSurfaceAnalyzer.CLI`.

Binaries

The application does not come with an installation program but binaries are provided with each release branch update for convenience and can be downloaded for immediate use or you may build the source code and run it. Pre-built binaries are located under releases.

Running ASA from the GUI

Startup

- Windows: open an Administrator Command Prompt or Powershell and run `Asa.exe gui`.
- Mac OS/Linux: use `sudo Asa gui`.

Operating

- Once you have started Attack Surface Analyzer it should automatically launch a browser window in your system default browser to `http://localhost:5000` with the application.

Collecting Data

- Select 'Scan' located from the top menu or 'Start Scan' from the home page. **Note**: Scanning should never be run on live production servers since it can severely degrade the performance of the system.
- There are two options for collection of data: 'Static' or 'Live'. Static is selected by default and will collect indicated information for analysis.
- For a Live snapshot enter the directory you want monitored and click 'Scan'.
- ASA will take a snapshot of your system state and store this information in a local SQLite file. This initial scan is called the baseline scan. Be sure to name note the date/time for future reference. You will see the Scan page update indicating collection status.

- Install your product or applications necessary to configure the machine, enabling as many options as possible. Be sure to include options that you perceive may increase the attack surface of the machine. The baseline and product scans are now available to be analyzed for results.

Analyzing Results

- Select the 'Results' menu from the top, then select the baseline and product scan to analyze and compare against. Collection elements that have results will be indicated. Use the collection filter to select what to view from the analysis. Use the 'More' button to see additional results on the right.
- Review the identified changes to determine impact, severity and policy for the same.
- You can either analyze the results on the computer where you generated your scan, export the results to a JSON file or copy the SQLite file to another computer for analysis.

Running ASA from the Command Line

The CLI version of the tool comes with built-in help using a help parameter i.e. run `asa` with no arguments which lists all top level argument options or "asa help" where is one of the value listed when arguments are supplied e.g. "asa help collect". Follow the same general baseline and product snapshot procedures for the GUI application.

Performance Considerations

Note that analyzer has high CPU and disk I/O demands, and may take a considerable amount of time to complete. Analyses should never be run on live production servers since it can severely degrade the performance of the system.

If you are scanning system components on Windows you may notice significantly degraded performance with Defender. This entry in the FAQ provides a potential workaround: https://github.com/microsoft/AttackSurfaceAnalyzer/wiki/FAQ's#windows-defender-is-consuming-a-lot-of-cpu-when-running-asa

System Requirements

Operating System Support

ASA is tested on Windows 10, Linux and MacOS systems. No installed pre-requisites or redistributables are required, beyond those of .NET Core.

Our core technologies are ASP.NET and .NET Core 3.1. No other systems are tested at present but .NET Core is supported on the following versions of Windows:

- Windows 7 SP1
- Windows 8.1
- Windows 10 Anniversary Update (version 1607) or later versions
- Windows Server 2008 R2 SP1 (Full Server or Server Core)
- Windows Server 2012 SP1 (Full Server or Server Core)
- Windows Server 2012 R2 (Full Server or Server Core)
- Windows Server 2016 or later versions (Full Server, Server Core, or Nano Server)

Additional OS compatibility for .NET Core is located here https://github.com/dotnet/core/blob/master/release-notes/3.1/3.1-supported-os.md.

For Users of Attack Surface Analyzer 1.0

Note that .cab files generated from versions of Attack Surface Analyzer are not compatible with Attack Surface Analyzer 2. You will need to run a new baseline and product scan to perform the analysis.

Support

For submitting defects, just use the standard GitHub Issues link.

Security issues and bugs should be reported privately, via email, to the Microsoft Security Response Center (MSRC) at secure@microsoft.com. You should receive a response within 24 hours. If for some reason you do not, please follow up via email to ensure we received your original message. Further information, including the MSRC PGP key, can be found in the Security TechCenter.