

[Home](#) > [Securing productivity applications](#) > [Network security](#) > [application whitelisting](#)

DEFINITION

application whitelisting

Posted by: [Margaret Rouse](#) [WhatIs.com](#)



Contributor(s): [Brien Posev](#) and [Peter Loshin](#)



SearchSecurity



whitelisting is to protect computers and networks from potentially harmful applications.

In general, a whitelist is an index of approved entities. In information security ([infosec](#)), whitelisting works best in centrally managed environments, where systems are subject to a consistent [workload](#). The National Institute of Standards and Technology ([NIST](#)) suggests using application whitelisting in high-risk environments, where it is vitally important that individual systems be secure and less important that software be usable without restrictions. To provide more flexibility, a whitelist may also index approved application components, such as software libraries, plugins, [extensions](#) and configuration files.

Application whitelisting vs. blacklisting

Unlike technologies that use [application blacklisting](#), which prevents undesirable programs from executing, whitelisting is more restrictive and allows only programming that has been explicitly permitted to run. There is no consensus among security experts over which technique -- blacklisting or whitelisting -- is better. Proponents of blacklisting argue application whitelisting is too complex and difficult to manage. Compiling the initial whitelist, for example, requires detailed information about all users' tasks and all the applications they need to perform those tasks. Maintaining the list is also demanding because of the increasing complexity and interconnections of business processes and applications.

Proponents of whitelisting argue it is worth the time and effort needed to proactively protect systems and prevent malicious or inappropriate programs from entering the network. Using a whitelist that allows only applications that have been explicitly approved offers more protection against [malicious software](#), rather than the looser standard used by application blacklists, which permit any software to run unless it has been discovered to be malicious and has been added to the blacklist.

How application whitelisting works

Implementation of application whitelisting begins with building a list of approved applications. The whitelist can be built into the [host](#) operating system (OS), or it can be provided by a third-party vendor. The simplest form of whitelisting allows the system administrator ([sys admin](#)) to specify file attributes associated with whitelisted applications, such as file name, file path and file size.

Windows AppLocker, which Microsoft added to [Windows 7](#) and [Windows Server 2008 R2](#), allows sys admins to specify which users or groups of users are permitted to -- or not permitted to -- run particular applications. In addition to restricting access to specific applications, AppLocker can be used to restrict users from installing new software, define which versions of a piece of software are permitted to be run and provide control for running licensed software.

Risks of using application whitelisting

Attackers can replace whitelisted applications with malicious apps with relative ease by creating a version of their malware that is the same size and has the same file name as a permitted application and then replacing the whitelisted application with the malicious one. Therefore, it is much more effective for application whitelisting software to use cryptographic [hashing](#) techniques coupled with [digital signatures](#) that are linked to the software developers.

Application control vs. application whitelisting

Although the terms are often used interchangeably, *application control* and *application whitelisting* are two different things. Both of these technologies are designed to prevent the execution of unauthorized applications. However, application control is not as stringent as true application whitelisting.

Application whitelisting is designed to monitor an OS in real time and prevent the execution of unauthorized files. This goes beyond simply preventing unwanted applications from running. Application whitelisting may also restrict the use of [PowerShell](#) scripts and other types of scripts in an effort to prevent [ransomware](#) attacks.

Although application control can be thought of as a form of application whitelisting, it is primarily designed as a tool for preventing unauthorized applications from being installed. When someone attempts to install a new application, the installation package is compared against a list of authorized applications. If the application is found to be authorized, then the installation process is allowed to continue.

While it is true that application control can be an effective tool for preventing the installation of unauthorized applications, the technology has two significant shortcomings. First, application control works at the installation package level, not at the file level. This means that it does nothing to prevent someone from running a stand-alone executable file or an application that is already installed on the system. This means that, while application control can be a useful tool for application management, it isn't particularly effective at preventing ransomware attacks.

The other major shortcoming associated with application control is that application control tools do not perform a granular inspection of application installation packages. When someone attempts to install an application, the application control mechanism only checks to see whether or not the application itself is allowed. If the application is authorized, then the application control tool assumes that the application installation package is trustworthy. It does not verify the integrity or authenticity of the files within the application installation package. As such, an attacker could conceivably install unauthorized code by slipping it into an otherwise legitimate application package.

Advantages

There are a number of benefits associated with using application whitelisting. It is worth noting, however, that some application whitelisting tools are more feature-rich than others and that not every tool delivers all of the benefits discussed in this section. For example, Microsoft's AppLocker, which is a part of the Windows OS, provides basic whitelisting capabilities but lacks the rich reporting and alerting capabilities that are commonly found in third-party solutions.

The best advantage to using application whitelisting is that it provides protection against ransomware attacks and other types of malware attacks. Traditional [antivirus software](#) tends to be signature-based. In other words, when a user attempts to launch an executable file, the antivirus software compares the file's hash against a database of code that is known to be malicious. If no match is found, then the file is allowed to execute.

In some ways, the use of antivirus software is similar to application blacklisting. The antivirus software explicitly forbids the execution of software that is known to be malicious. The problem with this approach, however, is that new malware is created every day, and it is impossible for any antivirus software application to maintain a completely comprehensive database of malicious code.

In contrast, application whitelisting is far more restrictive. It does not allow any executable code to run unless an administrator has explicitly granted approval. This greatly diminishes the chances of a ransomware attack or other malware infection occurring.

Depending on an application whitelisting tool's reporting capabilities, such a tool may help the organization to determine which users are engaging in risky behavior. Some application whitelisting tools are able to create reports detailing which users have attempted to install or run unauthorized applications, as well as any malware that has been detected.

Another benefit to using application whitelisting is that doing so can simplify [software license](#) compliance. To be fair, most application whitelisting tools are not designed to perform license metering. At the same time, however, restricting the use of unauthorized applications prevents situations in which an auditor flags the organization for a license violation as a result of someone using an unlicensed application that the IT department did not even know about.

One more potential benefit to using application whitelisting is decreased help desk costs.

Application whitelisting allows an organization's IT staff to not only restrict which applications users

are allowed to use, but also to control which versions of an approved application can be run. These restrictions have the potential to drive down help desk costs since they eliminate the possibility of users installing a piece of software that interferes with another application on the system. It also gives the IT staff the ability to make sure that users are running application versions that are known to be stable and reliable.

Uses

Because application whitelists can be tedious to configure and maintain, the technology is used primarily within organizations that demand the best possible security, as well as extremely tight control over application usage. An organization might, for instance, have contractual or compliance mandates that require specific applications to be used.

Although somewhat counterintuitive, application whitelisting has also been successfully used by small organizations. Small and medium-sized businesses ([SMBs](#)), by their very nature, tend to rely on a small and relatively static collection of applications, which makes application whitelisting relatively easy to deploy and maintain.

Implementation

The application whitelisting implementation process varies considerably depending on which whitelisting tool is being used. Regardless, there are several best practices that should be adhered to during the implementation process.

First, before an organization begins deploying the application whitelisting software, it is critically important to compile a comprehensive inventory of the applications that are used throughout the organization. Remember, all of these applications will need to be included in the company's whitelisting policy. The application whitelisting software is designed to enforce [endpoint security](#), so any software that is not explicitly listed within the policy that the company creates will not be allowed to run. This is why it is important to create a comprehensive inventory of the applications that the organization uses. Failure to identify an application and include it in the whitelisting policy will result in the application being made unavailable to users.

Another best practice is to be careful about how you define whitelisted applications. Some organizations choose to whitelist specific folders or file names. However, using this approach may make the organization vulnerable to ransomware attacks and other threats.

The problem with identifying applications by their files or by the folders that they use is that malware authors can easily create malicious code that uses the same file names or folders as legitimate applications, thereby fooling the application whitelisting software.

The best way to ensure good endpoint security is to identify applications by using the publisher's signature or by using a cryptographic file hash. Most application whitelisting tools will allow you to base your whitelisting policy around both of these identifiers.

A slightly less effective, but still viable technique is to identify applications based on the [registry keys](#) that they create. The main problem with building a whitelisting policy around a series of registry keys is that not all executable code utilizes the registry. Most PowerShell scripts, for example, do not create registry entries.

Similarly, building a whitelisting policy that is based primarily on registry keys can expose your organization to various threats to endpoint security simply because it is easy for a malware author to spoof a legitimate application's registry keys.

Management of application whitelisting

If an organization plans to use application whitelisting, it must consider how it will handle the long-term management of the whitelists. Any time that the organization adopts a new application, that application must be added to the whitelist policy before it can be used. Similarly, an organization typically cannot upgrade an existing application to a new version unless it first adds the new version to the whitelist.

[Margaret Rouse](#) asks:

What kind of application whitelisting technologies have you worked with, and how well have they performed?



[Join the Discussion](#)

The bigger challenge associated with application whitelisting is that of intertwining the application whitelist management and [patch management](#) processes. Unless an organization has a plan for dealing with the patch management process, application patches will cause the whitelisting software to cease to recognize the patched application as being legitimate.

Most organizations use Windows Server Update Services ([WSUS](#)) or a similar tool for patch management. These types of tools give administrators the chance to approve patches rather than simply allow endpoints to download patches automatically. As administrators approve a patch for deployment, they can also add the patch to the whitelist policy.

Another possible solution is to base the application whitelisting policy around vendor digital signatures. That way, if a vendor releases a patch, then the patch will automatically be approved for use because it contains the same digital signature as the application that it is updating.

One more possible solution is to look for a vendor that keeps up with patch releases on your behalf and automatically updates whitelists to reflect newly released patches. Of course, this approach might be slightly less desirable since the vendor may whitelist a patch that the organization does not wish to deploy.

This was last updated in November 2019

Continue Reading About application whitelisting

- [How application whitelisting can help prevent advanced malware attacks](#)
- [Using whitelisting technology to defend against POS malware](#)
- [How a hybrid whitelisting-blacklisting approach can help enterprises](#)
- [Blacklisting or whitelisting, which is better?](#)
- [Application whitelisting, the battles you can win](#)

Related Terms

sandbox (software testing and security)

A sandbox is an isolated testing environment that enables users to run programs or execute files without affecting the ...


[See complete definition](#) 

two-factor authentication (2FA)

Two-factor authentication (2FA) is a verification process in which the user provides two different authentication factors to ...

[See complete definition](#) 

Zoombombing

Zoombombing is a type of cyber-harassment in which an individual or a group of unwanted and uninvited users interrupt online ... [See complete definition](#) 

Dig Deeper on Productivity apps and messaging security

External collaboration in Webex Teams limited by new security control

By: Jonathan Dame

Strategies to mitigate cybersecurity incidents need holistic plans

By: Mike Chapple

10 endpoint security products to protect your business

By: Linda Rosencrance

Android vulnerability: How can users mitigate Janus malware?

By: Nick Lewis

Join the conversation

 12 comments

Create Username and Add My Comment

Margaret Rouse

- 10 Jun 2011 9:56 AM

What kind of application whitelisting technologies have you worked with, and how well have they performed?

Reply

Stevehill

- 30 Jan 2019 2:09 PM

Whitelisting the best way to secure your environment, ThreatLocker application whitelisting is simple to use and also has storage controls, which whitelists devices like USB drives and external storage devices.


Reply

[\[-\]](#) [Snogherjsk](#) 

- 21 Aug 2015 8:47 PM

How do you unwhitelist files?

[Reply](#)

[-] ncberns 

- 23 Aug 2015 7:16 AM

We've always used an ever-changing list of USE THIS, NOT THAT (or whitelist, blacklist if you prefer).

When we launch a new project, we expect our hire to follow the list. Since many arrive with installed programs, we test those while closely track their use. By the time the project wraps, the new programs will be on one list or the other....

[Reply](#)[-] **Peter Loshin**

- 17 Jan 2017 1:03 PM

@Snogherjsk: The answer to your question depends on the kind of file, and the way you are doing application whitelisting -- and I don't have specific expertise in doing this (maybe someone else has such an answer?).

However, that said, application whitelisting systems may offer some control over which types of files that are opened with particular applications, such as Word or Excel files.

Other types of files, such as configuration files or plugins or any other type that might be considered "executable" would also likely to be covered by controls provided by the application whitelisting system in use.

[Reply](#)[-] **Ihlediju**

- 22 Nov 2017 1:21 AM

In an environment in which employees have the latitude to download and install application for expedient purposes is well ripe for application whitelisting so as to prevent rogue application and limit infection and application vulnerabilities that can create a security risk for the organization.

[Reply](#)[-] **Peter Loshin**

- 22 Nov 2017 9:46 AM

Whitelisting is a great strategy for preventing information security incidents related to malicious (or otherwise unwanted) applications -- but whitelisting is just one such strategy.

By itself, whitelisting is not going to be a complete security solution.

And all organizations, even those that don't allow users to install any software at all, can still benefit from implementing whitelisting as a part of a fully in-depth security strategy.

[Reply](#)[-] **GeriB11**

- 31 May 2018 9:15 AM

Is whitelisting the domain the solution on Windows 10 to gain access to a secure domain?

Reply

[-] Peter Loshin



- 31 May 2018 9:44 AM

Maybe -- though I'm far from an expert on Windows 10.

Presumably, you already have access to that domain (if you don't, you probably shouldn't be trying to access it without permission), in which case (I guess?) it would work.

But, again, I'm not an expert...

Reply

[-] zigshanklin



- 13 Sep 2018 12:13 AM

Normally when in a room full of cybersecurity engineers I start with two questions:

- (1) Have you heard of Application Whitelisting (AWL)? They nod yes, but show their disdain.
- (2) Have you ever known someone to deploy it successfully? They generally burst out in laughter.

Only one person said they had some kind of success. Here's my experience ...

I've developed three cybersecurity products acquired by Cisco, Wheelgroup Corp., Netranger (NIDS/NIPS) and NetSonar (a network scanner), in 1998; and Psionic Software, First Response (an automated remediation system) in 2002.

My first experience with Application Whitelisting was when I took over as the VP of Engineering for Coretrace, an Application Whitelisting company, in Dec 2011, which was acquired by Lumension Security in Nov 2012 and re-licensed to Sophos in 2013. After my experience and eight cybersecurity patents for NIDS/NIPS, which use blacklisting technologies (no matter how clever they appear they're a default-allow strategy), Application Whitelisting (a default-deny strategy) made perfect sense. But what I found was that AWL in 2012 was very difficult to deploy and manage, especially in a network of heterogenous computers or OS images.

While I was able to clean up the Coretrace product before it was acquired by Lumension, it had a problem scaling beyond 300 to 500 endpoints because it, like many other AWL systems, have complicated rules and user interfaces for adding EXEs and DLLs to the whitelist. (Coretrace never got their execution control for scripts working. Their "learn mode" and "trusted change" were even worse. But the company was sold before I could implement architectural changes to address the fundamental problems of its usability.) It blocked everything (except scripts) the way it was intended, but it just wasn't manageable. Coretrace's Bouncer was not the only AWL product with management issues. One competitor tried to use a rating system that force you to access some unwanted Apps just to enable those you needed. With some AWL products,

you had to exclude entire directories from enforcement just so system could work with some Apps, a situation which essentially defeats the purposes of AWL once someone knows there's a hole in the armor.

In 2013, Steve Snapp and I launched White Cloud Security, which resolved the laundry list of architectural issues I'd identified in the Coretrace product. The first goal was to make it easy to deploy and manage, along with the ability to easily share the load of the whitelist management between trusted Admins. We also addressed several major security holes now apparent in traditional AWL, the SHattered Attack (which spoofs an App's SHA-1 identity) and malicious "lone-wolf" security admins who intentionally add unauthorized Apps to the trust-list.

We released the product in July 2015, and have added features to make it easier to use, including execution control for systems that use software that dynamically generate scripts that can never be added to the whitelist because they are different on each invocation. This is an important feature, because an Enterprise or SMB can't use a whitelisting technology if it can't handle every kind of App and Script that the business needs to run within their network.

Another important thing we did was to eliminate the "whitelist push" model in favor of a client-server model. This keeps the trust-list secure, while also eliminating the need to constantly push the whitelist out to all endpoints.

We've also implemented Trusted Scripts, which applies execution control to scripting languages, a feature missing in many AWL systems. If you can't control the scripts running on an endpoint, then you're not really controlling what runs on the endpoint.

It has been a hard sell to most IT people. Because of the black-eye they got when punched in the face by their first (and only) AWL deployment, most of them are unwilling to believe that there's a simple paradigm that dramatically reduces the Trust-Listing workload. But we find that once a user or organization is locked down, they rarely go back to relying upon blacklisting technologies alone. Those that do, often come back with two black-eyes and a broken nose from a ransomware infection that their blacklisting technology didn't stop.

We're always looking for feedback and enjoy answering questions about blacklisting vs whitelisting, because both interactions help us improve our product.

Reply

[-] frankstechlab



- 29 Jan 2019 3:56 PM

Reply

[-] 27longak



- 18 Apr 2019 9:18 AM

i like your post

Reply

[CLOUD SECURITY](#) [NETWORKING](#) [CIO](#) [ENTERPRISE DESKTOP](#) [CLOUD COMPUTING](#) [COMPUTER WEEKLY](#)



SearchCloudSecurity

How to pass the AWS Certified Security - Specialty exam

Author of 'AWS Certified Security - Specialty Exam Guide' Stuart Scott shares insights on how to prepare for the exam and reap ...

Practice AWS Certified Security - Specialty exam questions

Explore the security and compliance capabilities of the AWS Config service to prepare for the wide-ranging AWS Certified Security...

[About Us](#) [Meet The Editors](#) [Contact Us](#) [Videos](#) [Photo Stories](#) [Definitions](#)

[Guides](#) [Advertisers](#) [Business Partners](#) [Media Kit](#) [Corporate Site](#)

[Contributors](#) [CPE and CISSP Training](#) [Reprints](#) [Events](#) [E-Products](#)

All Rights Reserved,
Copyright 2000 - 2020, TechTarget

[Privacy Policy](#)

[Do Not Sell My Personal Info](#)

