What is Application Whitelisting?

Created by Dennis Underwood, last modified on Aug 22, 2022

Introduction

Application whitelisting, in Cyber Crucible's context, is giving an application permission to perform some action that otherwise might be considered suspicious or malicious.

Application whitelisting technology is typically found in behavioral analysis defensive tradecraft, though some attackers use it as well (not covered here) in their operations.

Think of it like at school, students were not normally allowed to walk the hallways during class. However, if they were given a "hall pass" in some form by an authority figure, students were not returned to their classroom or delivered to the principal's office.

Types of Application Whitelisting - a non-technical description

If we continue with the "hall pass" analogy, there can be different techniques that school administration officials have learned, which helped formalize the behaviors some students (definitely not our CEO Dennis) may have tried to get away with. These all have direct correlation to security operations.

The Obvious & Cumbersome Hall Pass (a good thing)

21987331.jpg?width=278
Ever remember getting a hall pass, or a key to some room, that was incredibly large and obvious?

There were a couple good reasons for that:

1. So it would not get lost.
2. To lower the time and energy investment from an authority figure, in quick verification of the student's approved activities.

Good software won't "lose" a whitelist, but application whitelist software should enable fast, resource-efficient validation that the application has been inspected.

The Generic Hall Pass (probably not a good thing)

This is probably the least useful to application monitoring and schools alike, but requires the least management or expertise.

In this case, the student is allowed to run their errand in the hallway. The teacher may know, for example, but there is no validation from hall monitors as to what the student is doing. Also, if there is attempted validation, there are two levels of considerable resource investment:

1. the level of effort to approach the student, and interrogate them as to their behaviors and intent
2. the level of effort to ensure the student is being honest, to validate the student's explanation

With applications, this is most often seen when a program is added to a whitelist, and there is no further validation of behaviors or allowed activities. Unfortunately, the ability to gain context to an application's behaviors are typically not available for acquisition if not tracked externally by some other framework.

The Generically Specific Hall Pass

Not to be confused with, "go ahead Application, do whatever you want!"

Let’s go back to the restroom pass, because it is so appropriate here. Given the pass' visibility, it is relatively easy to assess what the student should be doing. Egregious behaviors out of line with “going to the restroom” behaviors may be assessed without much difficulty.

Applications whitelisting which has generic boundaries of activities are the most common, though in many cases those boundaries are the equivalent of allowing a student with a restroom pass to play on the playground.

The Specific-Specific Hall Pass

Not to be confused by the “generically specific” hall pass!

22642689.jpg?width=340
While, with the generically-specific hall pass, we know the student is using the restroom - in a large school, we do not know
1. Who gave the student permission

2. Where they just came from.

3. Which restroom the authority figure had in mind when they gave approval.

4. How long the student has been “out and about”.

This required more work on the authority framework for the school.

To return to application whitelisting, behavioral analysis is more accurate with more variables added to any decision-making capability.

Additionally, while investigating a hall pass that has been issued, you normally do not need to trace back a chain of 5 teachers to get to the source. You also can almost always trust the teacher wasn’t some type of “bad teacher” that is telling students to behave improperly. Application whitelisting, in a specific-specific whitelist example, requires working backwards, to examine every parent or ancestor application that opened the next application. For example, a piece of malware may have opened up a legitimate Windows utility.

The following questions are roughly equivalent to the “student hall pass” above:

1. What was the application started for, or told to do? We need to know the intended behavior.
2. Who started the application? This is typically another program.
3. Which “parent” or “ancestor” programs resulted in this program opening? What were each of them doing, or intending to do?

The Compromised Application

Remember in various science fiction, or horror literature and movies, when an imposter has somehow taken

22413315.jpg?width=226
over control of a character’s behavior? In schools, there isn’t much chance of an instructor being taken over by aliens.

Application takeover is a favorite technique of attackers because, like in the movies, most observers see nothing amiss until it is too late.

In this case, an attacker takes a running program, that is currently loaded into memory. So, the fully intact program in the file system is ignored. The program in memory is edited to add the attacker’s code. Much like an alien inside a human’s body. The attacker does their criminal behaviors, and when the program is done running, most of the evidence is gone. When the legitimate program runs again, if the attacker does not edit the code again, it is just as the original legitimate coders intended. Maybe a better analogy, in that case, would be a werewolf?

An application whitelist, then, needs extra work to ensure that the program, and any ancestors or parent programs, are intact without hacker editing the programs while they are running.

In Summary - Questions to Ask When Investigating an Application Whitelist Capability

Make sure to understand the boundaries of a whitelist allows, when discussing application whitelisting with your technology team or vendor.

1. Does this capability examine and track what a program is supposed to done, and compare expected vs observed behaviors?
2. How much detail does the whitelist capability capture in its decision making? More, or less, than a detailed student hall pass?
3. Does the whitelist capability track and investigate parent and ancestor programs for maliciousness? (Program A opens B, opens C, opens D – very common)
4. Does the whitelist capability validate the running program has not been tampered with?

Deep Reading from the National Institute of Standards & Technology (NIST)

An authoritative publication that is not for casual reading, but is very descriptive.

While it does not cover every scenario, and specifics are not updated frequently enough to keep up with every scenario, the descriptions are strategic enough to be valuable in most circumstances.

The Cyber Crucible team always values insight from NIST publications, and sets aside time accordingly.

https://nvlpubs.nist.gov/nistpubs/specialpublications/nist.sp.800-167.pdf

---

Revision #2
Created 2026-05-18 20:28:13 UTC by Dennis Underwood
Updated 2026-05-18 20:39:48 UTC by Dennis Underwood