



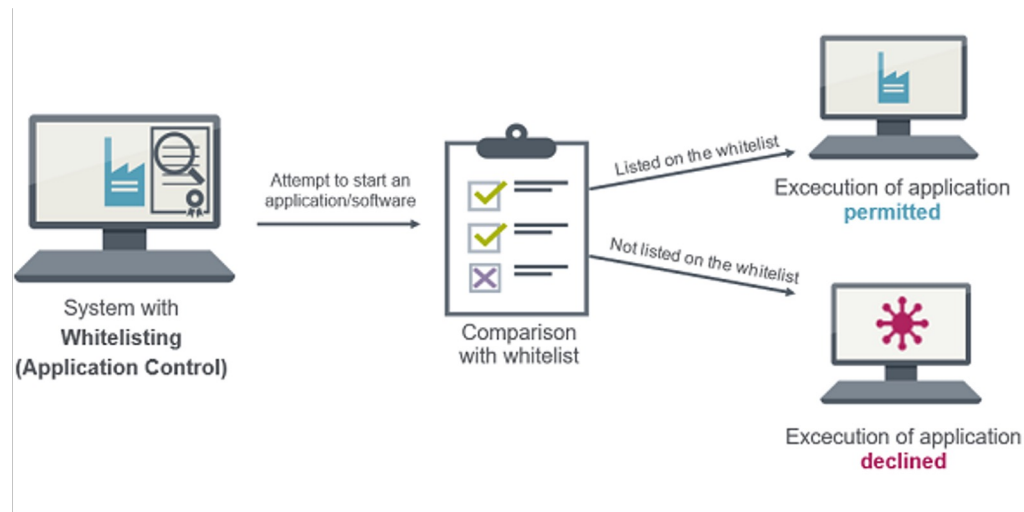
# Application Whitelisting

*By Palmer, Pierre and Tung, Matthew*



# What is application whitelisting

- List of application(s) and application component(s) (libraries, configuration files, etc.) that are authorized to be present or active on a host according to a well-defined baseline. (whitelist, allowlist, passlist)
- List of application/components authorized when Everything is Denied BY Default.



[1]



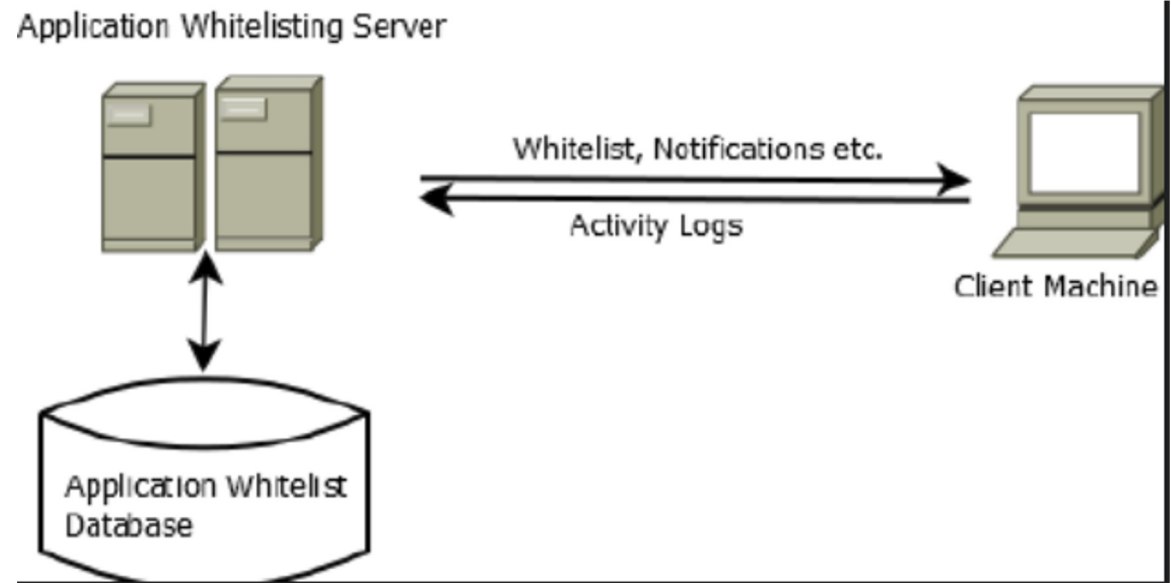
## Whitelisting Modes

- Audit mode: continuous monitoring process in analyzing the logs of the executions, including those not on the whitelist.
- Enforcement mode: Either permits or blocks items on the whitelist.
  - ★ Whitelist enforcement: all items are allowed to execute that have been whitelisted.
  - ★ User prompting: requests permission from either the user or \*administrator.
  - ★ Blacklist enforcement: blocks what is not listed specifically on the whitelist.





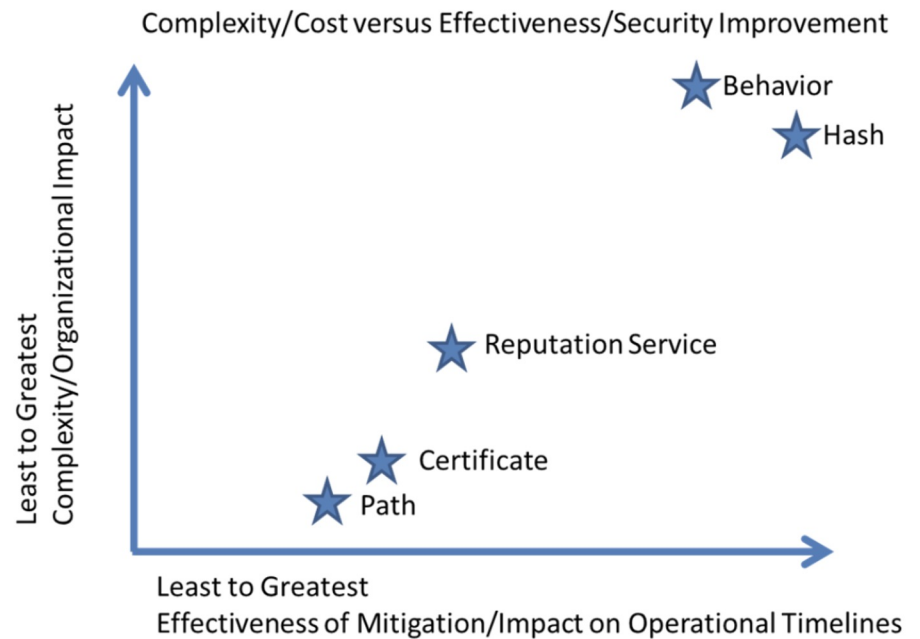
- Reduces execution of malware and unlicensed software.
- Zero-trust access to whitelist applications and tools.
- \*Only administrators shall be allowed to grant access if not on whitelist.





# Types of application whitelisting

- File path / file name
- File size
- Digital signature
- Cryptographic hash
- Reputation-based
- Behavior-based



[1]



## How are whitelists generated?

- Option 1) Vendor builds their own application list
  - Pros: each application vetted by the vendor, in theory should be no false negatives
  - Cons: vendor has to acquire patch information, can be problematic for high frequency patching
- Option 2) Scan files on clean host to build baseline
- Option 3) Vendor makes use of reputation services to make a decision on trustworthiness of a publisher.



## Uses for application whitelisting

- main one: **application access control**
- **software inventory**: identify unauthorized applications as well as incorrect versions
- **file integrity monitoring**: some software can actively prevent changes while others only report
- **incident response**: whitelisting technologies can check hosts for malicious files after one has been identified
- software reputation services
- integration with anti-malware technology



## Important design points

- Cryptography:
  - used to generate and verify cryptographic hashes for files
  - validates digital signatures for files
  - protects confidentiality and integrity of communications, such as lists of installed apps and versions
- Federal agencies are required to use FIPS or NIST-recommended algorithms contained in validated cryptographic modules
- Most solutions can only operate as centrally managed, and each end device must have software to enforce and audit
- Organizations must perform whitelist management, which directly affects how effective the solution is
- Testing should be done on lab devices to check not only correctness but logging/alerts, performance, and the whitelisting technology itself.





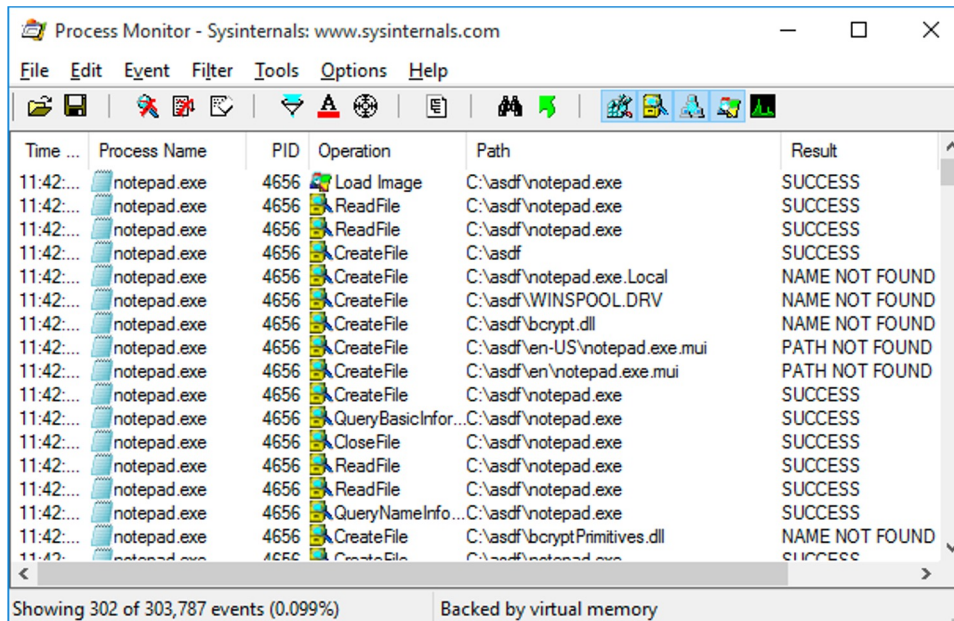
## Whitelisting limitations and risks

- whitelists may not be able to handle patched applications, especially those patched on an automatic or frequent basis. This may result in business interruptions or outdated software being allowed to run.
- whitelists do not protect against:
  - vulnerabilities in whitelisted applications themselves
  - malicious scripts that may be run (for instance from a whitelisted interpreter, or Microsoft Office macros)
  - memory-based attacks (fileless malware)



## Whitelisting limitations and risks (continued)

- commercial whitelisting software has been bypassed rather easily in numerous well-documented ways (through JavaScript, default whitelisted exe files, etc.)
- vendors may not be completely honest about what the software does and does not do

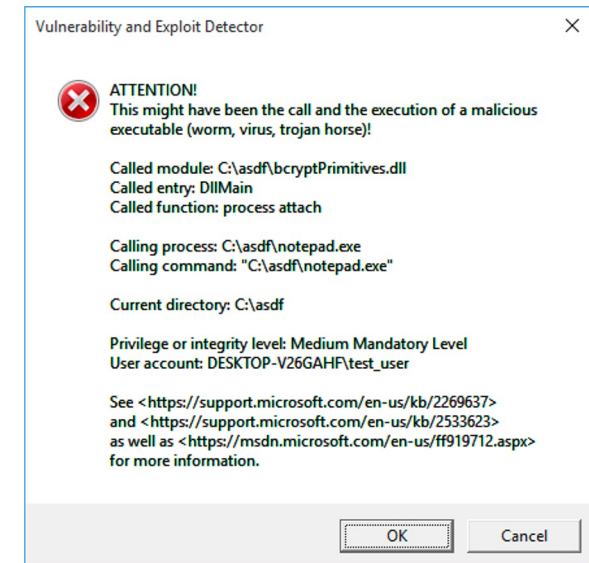
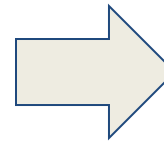


Process Monitor - Sysinternals: www.sysinternals.com

File Edit Event Filter Tools Options Help

Time ...	Process Name	PID	Operation	Path	Result
11:42:...	notepad.exe	4656	Load Image	C:\asdf\notepad.exe	SUCCESS
11:42:...	notepad.exe	4656	ReadFile	C:\asdf\notepad.exe	SUCCESS
11:42:...	notepad.exe	4656	ReadFile	C:\asdf\notepad.exe	SUCCESS
11:42:...	notepad.exe	4656	CreateFile	C:\asdf	SUCCESS
11:42:...	notepad.exe	4656	CreateFile	C:\asdf\notepad.exe.Local	NAME NOT FOUND
11:42:...	notepad.exe	4656	CreateFile	C:\asdf\WINSPOOL.DRV	NAME NOT FOUND
11:42:...	notepad.exe	4656	CreateFile	C:\asdf\bcrypt.dll	NAME NOT FOUND
11:42:...	notepad.exe	4656	CreateFile	C:\asdf\en-US\notepad.exe.mui	PATH NOT FOUND
11:42:...	notepad.exe	4656	CreateFile	C:\asdf\en\notepad.exe.mui	PATH NOT FOUND
11:42:...	notepad.exe	4656	CreateFile	C:\asdf\notepad.exe	SUCCESS
11:42:...	notepad.exe	4656	QueryBasicInfor...	C:\asdf\notepad.exe	SUCCESS
11:42:...	notepad.exe	4656	CloseFile	C:\asdf\notepad.exe	SUCCESS
11:42:...	notepad.exe	4656	ReadFile	C:\asdf\notepad.exe	SUCCESS
11:42:...	notepad.exe	4656	ReadFile	C:\asdf\notepad.exe	SUCCESS
11:42:...	notepad.exe	4656	QueryNameInfo...	C:\asdf\notepad.exe	SUCCESS
11:42:...	notepad.exe	4656	CreateFile	C:\asdf\bcryptPrimitives.dll	NAME NOT FOUND
11:42:...	notepad.exe	4656	CreateFile	C:\asdf\notepad.exe	SUCCESS

Showing 302 of 303,787 events (0.099%)      Backed by virtual memory





# Whitelisting from a Reference Monitor viewpoint

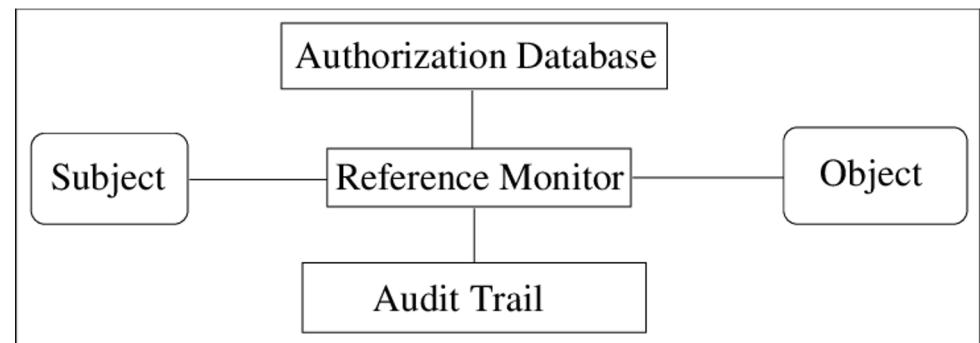
- Tamper proof: no
- Non-bypassable: no
- Verifiable: no

Subjects: users accessing applications

Objects: application or executable

Audit trail: built into many whitelisting solutions

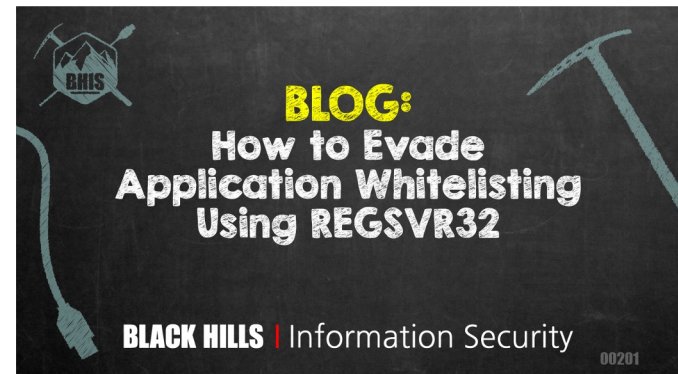
Authorization database: implementation depends on whitelisting solution





## Bypassing application whitelisting examples

- McAfee Application Control software: arbitrary code can run via hta and js files, which were simply unblocked by the software
- code can also be run via PowerShell, which is whitelisted by default
- AppLocker: command-line utility in Windows, Regsvr32 can be made to point to a remotely hosted file that can contain malicious scripts
- Bit9: was hacked and the company's signing certificates was accessed by the attackers.  
This led to them signing and spreading malware.





## References:

- [1][https://cache.industry.siemens.com/dl/files/603/109762603/img\\_256473/v1/AppleWhite\\_EN1.png](https://cache.industry.siemens.com/dl/files/603/109762603/img_256473/v1/AppleWhite_EN1.png)
- NIST Special Publication 800-167
- image:[https://www.pikon.com/wp-content/uploads/2020/12/ICON\\_VAT\\_Poland\\_-\\_White\\_List\\_Check\\_for\\_SAP.png](https://www.pikon.com/wp-content/uploads/2020/12/ICON_VAT_Poland_-_White_List_Check_for_SAP.png)
- image:<https://www.researchgate.net/profile/Himanshu-Pareek/publication/235981426/figure/fig2/AS:299798415593473@1448488920239/General-Architecture-for-Centralized-Application-Whitelisting.png>
- <https://insights.sei.cmu.edu/blog/bypassing-application-whitelisting/>
- <https://sansorg.egnyte.com/dl/yVZoYRvZX2>
- [https://prime.sba-research.org/wp-content/uploads/2016/09/sbaPRIME\\_WP\\_Application-Whitelisting.pdf](https://prime.sba-research.org/wp-content/uploads/2016/09/sbaPRIME_WP_Application-Whitelisting.pdf)