

Application Whitelisting - Extend your Security Arsenal?

Mike Baldi – Cyber Security Architect
Honeywell Process Solutions

Agenda

- What is Application Whitelisting (AWL)
- Protection provided by Application Whitelisting
- Optional features available with Application Whitelisting
- Technical description of Application Whitelisting
- AWL on Industrial Control Systems



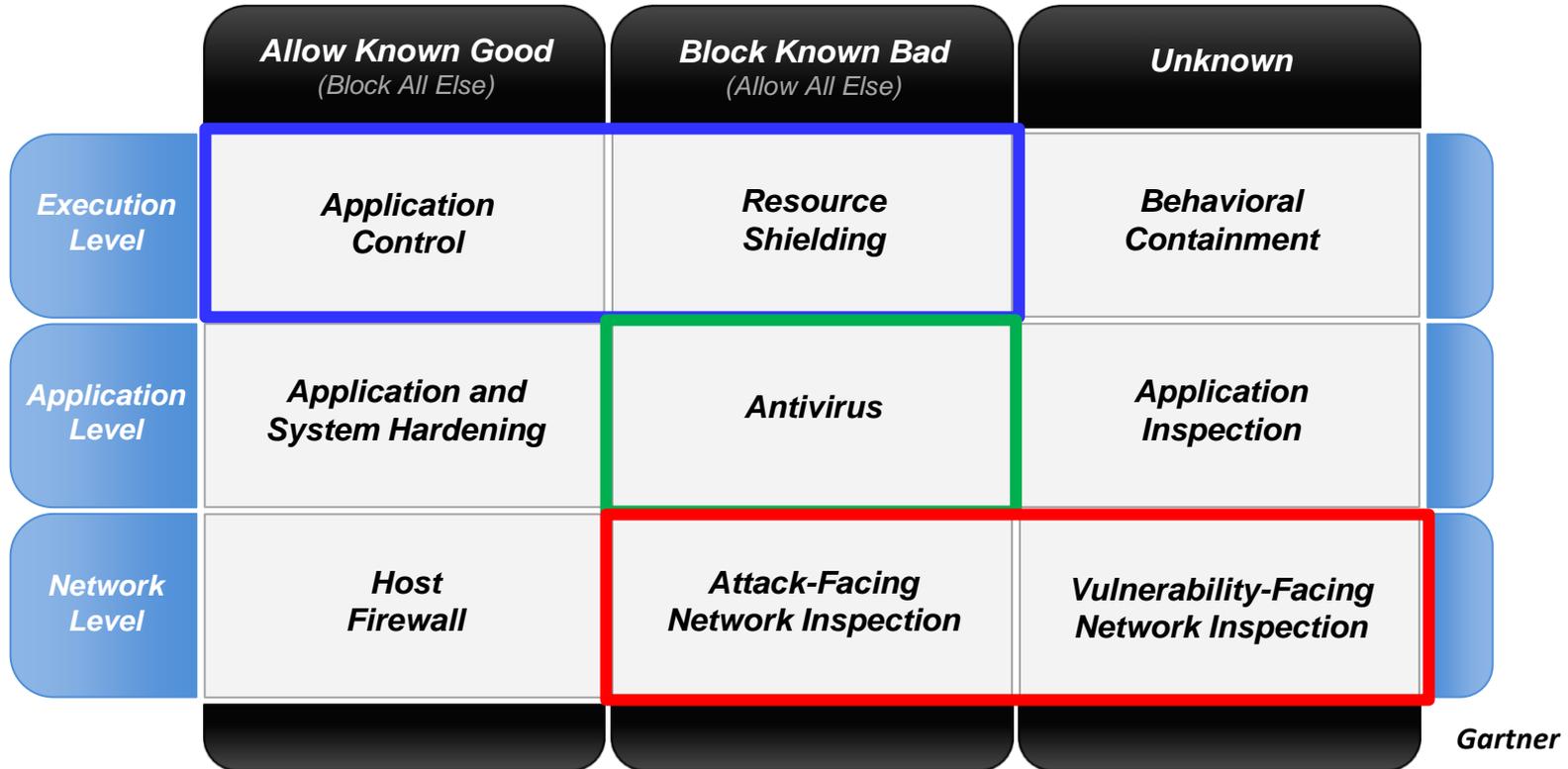
Some qualifiers...

- There are several GOOD Application Whitelisting (AWL) solutions available today
- Honeywell has qualified a specific vendor solution for use on our systems – and are evaluating an alternative solution
- Some of the technical details will focus on a particular vendor's implementation.
- The majority of this presentation will focus on generic AWL issues, and the impact on Industrial Control Systems

What is Application Whitelisting

- Application Whitelisting (AWL) is a technology which provides ***node-level*** protection against malware.
- Traditional ***Antivirus*** Software uses signatures of known malware to block “known bad files”
 - Only blocks on “known malware”, very reactive, constant updates required
 - Can result in false positives
- Application ***Whitelisting*** Operates on the principle of “only allow known good files to run”
 - Proactive, only needs updating when software changes are made to a system
 - Can result in false negatives (block something which should run)

Where does Application Whitelisting fit?



BL – Black Listing



AWL – Application White Listing



VP – Virtual Patching

How does Whitelisting provide protection?

- *It scans the hard disk for all executable code and creates a hash of all files it detects*
 - *Multiple hash codes are created (3 hash codes – MD5, SHA-1, SHA-256 for solution Honeywell implemented)*
- Files have to be “approved” for execution
- Rules for what to check, and how to react are assembled into a “policy”
- *Prior to starting an executable it checks:*
 - If the hash code exists it will check for any applicable rule in the active policy
 - *Will block execution of any file which has not been “approved”*

AWL Policies and Rules

- Policies are sets of rules which govern how AWL runs on each node
 - Honeywell provides a set of policies which covers all node types
 - Scripts and Database provided to pre-configure policies / setup for Honeywell DCS
 - One policy active per end-node at any given time
 - Each policy has a “setcon” value – defines security mode (lockout, monitor,...)
 - “default policies” provided by vendor - ie: for Microsoft, McAfee as updaters
 - Should have default policy on each end node – avoid “locking yourself out”
- Several types of Rules which make up Policies
 - File Integrity Control - prevents/reports changes to files or folders
 - Trusted Paths – folder where file execution is always allowed
 - Execution Control – tighter control on executing a specific file
 - File Creation Control – controls behavior when attempting to write a file
 - Performance Optimization – ignore creation, modification and deletion of files
 - { still monitoring / managing execution of the file }

What protection does Application Whitelisting provide?

- Protection provided by Application whitelisting
 - It protects all executables that are initiated / executed by MS Windows
 - Protects against Trojans
 - Malicious programs, or DLL files (DLL load order attacks)
 - ex; new version of notepad.exe with malware added to system
 - Modification of DLL or system library files with malware
- What AWL does NOT protect against
 - Memory based attacks
 - Inline DLL injection , inline IAT hooking
 - Interpreted code (ie: JavaScript, Pearl, web-based apps)
- For example malware like Conficker, Duqu (proof of concept) were not detected during tests.
- Did not protect against Alternate Data Streams in version tested



Optional AWL features – beyond “Basic AWL”

- Protection for USB devices
 - Lock down by operation type (read only, read/write, execute only...)
 - Enable only specific devices by vendor or serial #
 - Prompt before permitting USB operation
- Memory based attacks
 - Create custom rules to match profile of known memory attacks
 - Is risky, reactive solution – must be used with caution
 - AWL vendors are working on additional memory-based attack protections
 - Some implementations use memory mgmt techniques similar to DEP or ASDP
- Registry protection
 - Lock down registry hives or individual entries - by operation, location
- Enhanced file and folder access control
 - Prevent write access to folder or file (any file type, including text files)

More optional AWL features

- Snapshots of all files in the system
 - Baseline or gold configuration
 - Inventory of files on system
- Tracking every execution of file(s) in the system
- Lock down where a file can be executed from
- Drift reports available to track changes
 - Drift of all computers since initialization – or daily drift
 - Can take snapshot and compare with another end node
- **Global Software Registration service**
 - Offsite master file hash database of known good files
 - Requires internet connection

Some AWL technical details



Some AWL technical details

- AWL uses a server / client architecture
 - Server
 - manages multiple “agents” (end nodes / clients)
 - Console to interact with agents, configure policies, evaluate status, view reports
 - Database for policies and events
 - Collects AWL events and logs
 - End node
 - provides AWL functionality – governed by policy downloaded from Server
- What if clients lose communication with AWL Server?
 - Limited policy changes available via command prompt on End Node
 - End Node falls to default policy if unable to contact AWL Server on boot up
- Modes of AWL operation
 - Disabled, Monitor, Block-and-Ask, Locked-down

More AWL technical details

- AWL runs at kernel OS level of end node – and is active for every read and write
- AWL operation during file reads
 - Running at kernel level of OS – checks file when opened for read
 - If it's an executable or "file of interest " – verifies that it's approved
 - Checks rules in active policy before permitting operation
 - **Enforcement is at the start of execution**
- AWL operation during file writes
 - Running at kernel level of OS – checks file after write to disk
 - If file is an EXE or "of interest" type file – recalculates the HASH
 - New file is marked as "pending" if HASH is unknown / not approved
 - Recalculating the HASH codes occurs only when file is closed on disk write
 - Can't run "pending files" – depending on active AWL policy – considered "dirty"
 - Doesn't check writes to memory-resident files

AWL key considerations

- Protecting AWL on server and end nodes / agents
 - Tamper protection on end nodes and on server
 - Protect policies in SQL Server on server
 - Protection from unauthorized access to API
- Web server used by AWL to communicate with end nodes:
 - Apache Tomcat or IIS Server
- Change Management considerations
 - Certificate-based installation
 - Trusted installer (user ID based)
 - Trusted location
 - Manual approval
 - Temporary Policy for update

AWL considerations for ICS's

- Each agent – has an adjustable level of logging.
 - Disk space and cache rollover considerations
- Network load considerations:
 - AWL events uploaded from end nodes to server
 - Simultaneous startup of multiple end nodes
 - agents reconnecting to the server after an outage (uploading cache)
 - Logging level can overload network
- AWL must integrate into other security protections / tools (SIEM, RBAC, security Dashboard)
- What happens if a currently – used policy is deleted from the server? (ie: end nodes still running on this policy?)

Additional AWL considerations for ICS's

- Use same AWL solution for DCS and enterprise?
- What tools are available to rollout policy changes to entire system?
- Do you want a single generic policy to cover all ICS nodes, or specific policies for each node?
- Use a separate AWL Server, or install on an existing node?
- Virtual or physical AWL Server

AWL challenges on an ICS

- AWL has it's own “update and patch” cycle – has to be accommodated / incorporated into site practices
- How do you establish a “clean and trusted” system? (with no malware, including in boot areas of disk?)
- Notification of blocked file execution is via Windows Event Log
- Impact of pushing down new AWL policies during plant operation
- Policy changes when AWL server is unavailable
- Complexity / sequencing / quantity of rules can impact performance

AWL must be tightly coupled with ICS

- Must be “tuned” for compatibility with other security protection technologies – ie: anti-virus, EMT
- Consider performance problems – ie: setting logging level too high impacts system operation or chews up disk or network
- Must open up specific ports on the Host firewalls for AWL
- Must copy “trusted installer” certificates from certificate store to AWL Server database
- Must verify AWL doesn’t block critical functionality, or cause timing impact on critical functions

AWL Summary

- Application Whitelisting is simple in concept –
 - Protects individual nodes from malware by only permitting “approved” files to run
- It should be used in addition to Anti Virus protection
- AWL offers additional features beyond basic file execution protection
- It is NOT a set-and-forget or low maintenance solution
- If not properly configured and managed, it can lock out critical functionality
- primary challenge is locking down the system as tight as possible for maximum security, without blocking critical functionality
- Application Whitelisting has to be tightly integrated into an Industrial Control System, and thoroughly tested prior to deployment

Questions

