Various solutions protect endpoints against sophisticated malware by controlling applications. To help security architects understand the pros and cons of using application control for malware protection, this document assesses three technologies and their impact on users and IT operations.

### Key Findings

- Two application control technologies are raising the bar for unknown malware on endpoints:
  - **"Default-deny" application whitelisting** executes only "known good" applications. It is one of the strongest anti-malware controls for all types of endpoints.
  - **Application isolation** controls the access of an application. This technique is typically used to limit the impact of unknown applications until they are proven benign.
- Removing administrative rights from all users dramatically reduces the risk of malware. Privilege management solutions can remove local administrative rights from all users, even if some applications require administrative privileges.
- To implement application control across all endpoints, organizations must strike a balance between security (strict policies), user impact and operational overhead. In practice, this means categorizing endpoints and differentiating policies across endpoint categories.
- Successful implementations may require extensive planning, tuning, training (IT and user awareness) and policy design, but could require little operational effort after deployment.

### Recommendations

- Remove local administrative rights from almost all users. Privilege management solutions can help with policy-based elevation and exception management.
Categorize endpoints based on your requirements for security and for application installation flexibility. Define policies per endpoint category, balancing security with user impact and operational impact.

Complement application control with technologies that protect against pure in-memory and data-driven attacks, that compensate for lenient policies (monitoring, blacklisting and self-authorization) on dynamic endpoints, and that detect advanced attacks.

Table of Contents

Analysis ..................................................................................................................................................3
Application Control Versus Blacklisting for Malware Protection ..........................................................4
Objectives of Application Control ......................................................................................................5
Application Control Concepts ...........................................................................................................6
Categorizing Applications ..................................................................................................................8
  Known Good Files ..........................................................................................................................8
  Known Bad Files ............................................................................................................................10
  Gray Files .....................................................................................................................................10
Limitations of Application Control ....................................................................................................11
  Known Good Files Can Become Malicious ....................................................................................12
  Application Control Does Not Prevent Changes to Configuration Settings ................................13
  Application Control Depends on Successful Policy and Exception Management ......................13
Postexecution Application Control Through Isolation, Monitoring and Journaling .........................13
User Privileges Versus Application Privileges ................................................................................15
Application Control Solutions ........................................................................................................16
The Impact of Application Control on Users and IT Operations ........................................................19
  Limiting the Impact on Users ........................................................................................................19
  Limiting the Impact on IT Operations ...........................................................................................20
  Making Application Control Work ...............................................................................................21
Future Developments ........................................................................................................................22
Strengths ........................................................................................................................................23
Weaknesses ......................................................................................................................................23
Guidance ..........................................................................................................................................23
The Details ........................................................................................................................................25
Gartner Recommended Reading .........................................................................................................33

List of Tables
Analysis

Various solutions protect against old, new and even sophisticated malware by controlling applications. The scope of this assessment covers three technologies: application whitelisting, application isolation and privilege management. The first two technologies focus on controlling the application itself, while the last one focuses on controlling the user of the application:

- **Application whitelisting.** In this technique, only known good applications are placed on an approved list and allowed to run.
- **Application isolation.** This approach restricts what an application is allowed to access (for example, files, registries or network resources).
- **Privilege management.** In this approach, most applications run with standard user privileges. Applications run with administrative rights only on an exception basis, as permitted by policy.

Out of scope are endpoint solutions that use whitelists for reasons other than controlling application execution. Readers should note that lists of known software are used ubiquitously by many endpoint protection platforms (EPPs) and next-generation security solutions for optimizing malware scanning or reducing false-positive ratios (FPRs).

This assessment focuses on (1) the strengths and weaknesses of using application control for malware protection and (2) the impact of application control solutions on users and IT operations. In it, we explain what is required to make application control work. As described in our analysis and illustrated in Figure 1, aligning policy with flexibility and security requirements is key to achieving a secure, user-accepted and manageable application control environment. (See the Making Application Control Work section for a full explanation of the choices made in Figure 1.)
Application Control Versus Blacklisting for Malware Protection

The most common endpoint technology to protect against malware is blacklisting, whereby malicious code is identified (most commonly through signatures and heuristics), consequently denied execution, and removed from endpoints. However, most malware is highly polymorphic — in many cases, the code used during initial infections is even unique to the affected organization. Thus, the blacklisting approach often fails. Blacklisting solutions require a significant and increasingly unscalable R&D effort to identify these enormous numbers of malicious files.

Application control technologies rely on an opposite paradigm (see Figure 2). They are based on the fact that the vast majority of files on an endpoint are not only known, but known good. By contrast, "known bad" files, which are the focus of blacklisting, are exceptions on an endpoint. Unknown files, aka "gray" or suspicious files, are the most challenging. These files are neither known good nor known bad. Depending on how well a solution categorizes files, the percentage of gray files on an endpoint may range anywhere from a few percent to 30%.
For a comparison between application control and other endpoint malware protection technologies, see "Comparing Endpoint Technologies for Malware Protection."

Objectives of Application Control

The objective of application control is to prevent attacks by providing visibility and control over application execution.

Application control solutions provide very strong malware protection by following a simple mantra: Run only what is both required and trustworthy, and run it with the lowest possible privilege and access.

In contrast to anti-malware solutions, application control solutions assume everything is bad, or at least suspicious, unless proven otherwise. When organizations deploy application control correctly and take care to mitigate residual risks (see the Limitations of Application Control section), it is one of the strongest endpoint security controls.

Application control is widely believed to be a very strong malware control. The Center for Internet Security (CIS) mentions it as Critical Security Control 2 (CSC 2) in "The CIS Critical Security Controls for Effective Cyber Defense Version 6.0." The Australian Signals Directorate includes it as one of the "Top 4 Strategies to Mitigate Targeted Cyber Intrusions." Application control’s strength as an endpoint security technology is also widely published in Gartner research. For example, "Market Guide for Cloud Workload Protection Platforms" states, "... use application control and whitelisting as your primary server protection strategy."
Because application control provides great insight into, and control of, application use, it offers various non-security benefits as well. For example, application control can facilitate software asset management and reduce the likelihood of interoperability issues (caused by people installing new add-ons, new versions or new client applications). In addition, application control reduces the variation of endpoint images, which can considerably reduce operational costs.

All of this may seem too good to be true, and to some extent, it is. Achieving all the objectives of application control is still not a panacea. Moreover, various application control deployments fail outright, or they end up providing only visibility, without true policy enforcement. In most cases, the problem is not the technology, but rather the end-user impact and the IT operational challenges. To end up in a low-maintenance and highly secure state after rollout, successful organizations spend adequate time on planning and deployment.

**Application Control Concepts**

Application control solutions do exactly that: They control the execution of application code on endpoints. They can decide not only *whether* a file may execute, but also *how* that file may execute (that is, with what privileges and with what access to the endpoint’s resources).

Moving from left to right, Figure 3 provides a high-level view of how an application control solution works. An execution event triggers a look-up to a service (see the Categorizing Applications section, below), which leads to one of the following results:

- **File is known bad.** In this case, the file is denied execution and deleted.

- **File is known good.** In most solutions, this means that the file is allowed to execute freely. Some solutions do not believe in the concept of known good. Such solutions will contain, monitor and journal changes of even good files, just in case those files turn bad over time.

- **File is unknown.** Policy determines what happens to such files. In default-deny whitelisting scenarios, these files are denied execution. Solutions that support application isolation may strongly contain the execution and limit the file’s access to endpoint and network resources.

Applications that are permitted to run may be subject to post-execution control. With this optional functionality of application control solutions, the agent may restrict the application’s access to resources, monitor its behavior for malicious activity and journal any changes for quick remediation.
Figure 3. Application Control Concepts

Source: Gartner (July 2016)
Categorizing Applications

A critical component of any application control solution is the decision technology around, for example, the following questions:

- Is this file known?
- Is it good?
- Is it bad?
- Is it vulnerable?
- Does it behave maliciously?

This section focuses on the technologies that are used for such determinations.

Known Good Files

Most solutions have two stages for managing known good files: initial discovery and life cycle management.

The basis for known good files — that is, the initial list of known good files — is usually determined by a combination of the following techniques:

- **File inventories.** The application control solution assumes files are good at the time of deployment and builds an inventory of known good files — aka a whitelist. Solutions take different approaches to building whitelists:
  - **Per endpoint.** Some solutions build a unique whitelist per device, or even per user and device. This approach reduces the opportunities for malware to spread laterally, but it also increases management complexity.
  - **From a small set of baseline images, aka "golden images."** In heavily managed environments, endpoint images are based on golden images. Application control solutions can build an inventory from these golden images as the initial list of known good files.
  - **Per organization.** Application control solutions with the main objective of providing default-deny whitelisting centralize application inventories across all endpoints. This approach provides optimal visibility across all files in an organization.
- **Digital signatures.** The application control solution assumes that files signed by a specific vendor or publisher, such as Microsoft or Oracle, are benign.
- **File owners.** The application control solution assumes that files installed by a specific user, or by a member of a specific user group, are benign. This approach allows, for example, automatic whitelisting of centrally deployed software, but does not allow local installation by generic users.
- **File locations.** The application control solution assumes that all files in a specific directory are benign. This approach simplifies software deployment in some organizations, and it allows for lightweight exception management. For example, some organizations allow developers to drop
the files they need into a directory, which consequently triggers the whitelisting of such files. This approach is not necessarily the most secure, but in certain use cases, usability trumps security. Note that solutions implement this method differently. Some solutions restrict execution to only the files located within the specified directory. Others add the discovered files to a whitelist, allowing the execution of identical copies of these files, even outside the specified directory.

- **Global reputation lists.** The application control solution assumes that files with a trust level above a certain threshold are benign. The solutions usually depend on a vendor-supplied cloud service to provide this intelligence. Some solutions rely on publicly available lists, such as the National Institute of Standards and Technology's (NIST's) National Software Reference Library (NSRL).

Once a basis for a whitelist of benign files has been established, the whitelist needs to be managed. As is clear from the list above, all sources for known good files, with the exception of file inventories, are dynamic in nature. Because these sources of known good files update automatically, they require very little maintenance from the IT operations team.

In addition to supporting the dynamic whitelist sources mentioned above, solutions typically support a combination of the following methods for whitelist life cycle management:

- **Trusted changers.** The whitelists are automatically revised by designated trusted changers. These typically include trusted accounts and applications (such as system management software used to deploy new endpoint software), digital signatures, and file locations.

- **IT operations.** Execution events for nonwhitelisted files trigger an IT operations workflow. The administrator then decides among the following options:
  - Block execution for this event
  - Block execution for all occurrences of this file in the installed base
  - Allow this file on this endpoint
  - Globally allow the execution of this file for all endpoints

- **Users.** When users attempt to execute a file that is not explicitly whitelisted or trusted, they receive a warning from the system. In order to run the file, users must first submit a business justification. Some solutions provide a trust rating for suspicious applications, in the hope that users will consider such information before making a decision.

For the most critical endpoints, security architects typically run an inventory of applications and allow only very few trusted changers (usually in the form of an existing system management tool). For more dynamic endpoints, more automation is required, and often, policies will include global reputation lists and additional life cycle management options. Figure 4 depicts the specific options for managing known good files and the effect of these options on IT operations and security.
Known Bad Files

In addition to providing information for whitelists, global reputation lists can help detect known bad files. Most application control solutions either integrate with VirusTotal (or similar global lists of known bad files) or use static antivirus signatures. Because these look-ups are based on static evaluation of a file, they can't replace a strong EPP.

To identify malicious files, organizations may supplement global reputation lists with internal reputation feeds. This approach can enable fast remediation of new malware. Some application control solutions, such as Intel Security, Kaspersky Lab and Trend Micro, categorize applications, enabling organizations to blacklist specific categories of files, such as all browsers except for the corporate one.

Static evaluation is too weak to be the sole solution for blacklisting malware, but it is good enough to block specific file categories or individual unwanted files.

Gray Files

The application control vendor’s strength in identifying known good and known bad files determines how much effort an IT operations team must devote to handling gray files. Policy and workflow determine the simplicity of making correct and efficient decisions between allowing a genuine new update to a popular browser component and executing a piece of malware with the same name. Wrong or slow management of gray files leads to compromised endpoints or to failed application control deployments.
Following are the four ways to handle gray files, listed in order from most secure to least secure:

- **Do not allow.** This is the most secure policy option, commonly referred to as default-deny whitelisting.

- **Attest before allow.** In this approach, the organization tests for maliciousness before allowing full access. Common methods to implement this approach are:
  - Automatically submit the file and/or the metadata to the vendor.
  - Automatically submit the file to an on-premises or cloud-based network sandbox (sometimes referred to as a detonate-and-deny or a verify-and-deny approach).
  - Rely on IT operations. Once the gray file triggers a workflow, a member of the IT operations team analyzes the exception request and makes an informed decision.

Some solutions can isolate gray files, or even known good files, during execution. Some even record changes made by certain processes, enabling later rollback if those processes turn out to be malicious. See the Postexecution Application Control Through Isolation, Monitoring and Journaling section for more information.

- **Rely on user.** User self-authorization with business justification is a manageable, yet low-security, way of dealing with gray files. It is especially suited for dynamic client endpoints. In these use cases, the business disruption of a default-deny approach may be too high, or the IT operations team may be unable to handle the high number of exception requests. To help users make the right decisions, some solutions provide certain information about the gray files, such as a global trust rating. However, do not expect users to pay much attention to this information. Although some organizations have noticed a drop in the number of user-driven exception requests, mandating a business justification is not enough. User awareness training must thoroughly illustrate the risks of making the wrong decisions in these scenarios.

- **Monitor only.** The organization allows the execution of gray files. It uses its chosen solution for application visibility and for file-based incident remediation, rather than for application control.

Do not trust IT operations teams or end users to make swift and security-conscious decisions without help. To make the right decisions, both IT operations teams and end users require all the relevant information on a file, including context. Contextual information answers questions such as:

- What is the origin of this file?
- Who wants to run it? (Regular user or local administrator?)
- Where is the user? (Off or on the network?)

**Limitations of Application Control**

With its tight control over application execution, application control seems to be almost a "silver bullet" for protecting against known, and unknown, malware. It’s not. This section discusses the main limitations of application control that security professionals must be aware of. Security
professionals must consciously accept these limitations, or compensate for them by using additional solutions.

**Known Good Files Can Become Malicious**

One limitation of application control is that a known good file — such as a whitelisted corporate version of Internet Explorer with the latest Java runtime and Adobe Flash plug-ins — can turn malicious. This can happen in three ways:

- **Exploitation.** The file has a vulnerability that is exploited, resulting in a benign, whitelisted application that executes malicious code.
- **Injection.** An attacker injects, for example, a dynamic link library (DLL) into a file’s process, making the file load malicious code into its memory.
- **Interpreted code.** The file itself allows the execution of code. Examples include Internet Explorer running JavaScript, Adobe plug-ins running Flash, and PowerShell.

Vulnerability handling is a point of differentiation among application control solutions. Should vulnerability affect trust? Should a file reputation automatically deteriorate over time as more and more vulnerabilities are disclosed? Some solutions, such as BeyondTrust, take these questions very seriously. By contrast, others don’t account for known vulnerabilities at all. Instead, they defer to outside solutions for vulnerability handling.

Pure application-whitelisting players may argue that threats to whitelisted applications are outside the scope of their solutions. However, Gartner believes that application control vendors should take memory control and exploit prevention into account. If your application control solution does not offer protection against the exploitation of whitelisted applications, look for complementary solutions in the form of exploit technique mitigation (for example, Malwarebytes Anti-Exploit, Microsoft Enhanced Mitigation Experience Toolkit [EMET] or Palo Alto Networks Traps).

Although many attacks, at some point, drop files onto the endpoint and attempt to run them, all phases of an attack can be executed without actually doing this. For example, ransomware attacks are leveraging JavaScript and social engineering to enable and execute Microsoft Office macros. These attack phases don’t require any new executables. All they need is a whitelisted Windows Script Host or Microsoft Office. Application control solutions must support a wide range of nonexecutable file types, script control mechanisms and interpreters to successfully block intrusions that leverage attacker-controlled data executed by a whitelisted application.

When evaluating solutions, prefer solutions with strong script control mechanisms, memory protection features, and application isolation capabilities.
Look for compensating controls in the form of exploit technique mitigation and memory protection, if the solution itself does not offer such protection for whitelisted applications.

**Application Control Does Not Prevent Changes to Configuration Settings**

Another limitation is that application control, by itself, does not prevent changes to configuration settings. For example, an attack on a web server could change the application server’s configuration files so that the machine serves malware to its visitors. In this scenario, no files are executed on the server, and no interpreted code or applications are changed.

Some application control solutions include integrity protection for configuration files and for critical parts of the file system and registry, but others do not. If your application control solution does not include such capabilities, compensate by deploying file integrity/change control solutions.

When evaluating solutions, prefer solutions with strong configuration control mechanisms.

Look for compensating controls in the form of file integrity monitoring, if the solution itself does not offer such protection.

**Application Control Depends on Successful Policy and Exception Management**

We cannot end this section before highlighting another limitation of application control, which is not technology-related, but process-related. Policy and exception management processes have a critical effect on the value that organizations receive from an application control solution. Policies that are too strict won’t be accepted by business leaders, users or even IT. Policies that are too lenient will render all the effort useless. Too much exception management will make the solution too expensive, yet too little will affect user acceptance and possibly lead to overapproval.

Understand that application control can achieve its goals only with a strong balance between security, usability and manageability. Process and policy issues seriously impact the effectiveness of application control. Users and IT operations teams may make mistakes managing policies and exceptions, especially in situations where insufficient context and threat information is provided at the time of decision making.

**Postexecution Application Control Through Isolation, Monitoring and Journaling**

Application control does not necessarily cease once a decision is made to run a program. Optionally, application control can extend to the execution phase by:
- **Restricting the access of the application.** The application control solution restricts access to specific resources, such as registry keys, files and folders, and network resources.

- **Running applications at "least privilege."** Running an application with different privileges from the user that executed it enables almost all users to work with standard user privileges. Thus, organizations can restrict privilege elevation to only those applications that require it.

- **Monitoring behavior during the application's execution.** Applications that are allowed to execute are monitored for malicious behavior. Some solutions do this only for gray applications, while others do it for all applications.

- **Recording system changes incurred during the application's execution.** Journaling enables organizations to roll back changes if an application turns out to be malicious at a later stage.

- **Monitoring sessions.** In addition to journaling changes made by the application, organizations can record user interactions that occur within the UI, through the command line and via keystrokes. This documentation can be used later for privileged attestations or for training.

Figure 5 illustrates postexecution application controls. It further identifies the different strengths of application isolation. Most application control solutions do not implement postexecution application control. Those that do typically implement access restrictions in a lightweight fashion, leveraging Microsoft Windows APIs. Stronger application isolation comes in the form of sandboxing from vendors such as Symantec and Thycotic. Stronger isolation will contain malware much better, but it may incur greater system overhead. For additional detection and protection, organizations can complement isolation with other postexecution controls, such as monitoring and journaling.

Containment solutions, such as Bromium and Invincia, use application isolation as their primary technology. They isolate processes that handle risky content. These processes include browser sessions, as well as editors that handle downloaded files or files from USB drives. Containment solutions differ from the application control solutions covered in this assessment in terms of focus: Containment solutions focus primarily on postexecution isolation without controlling the execution of gray files. Moreover, containment solutions often apply isolation to a fixed set of applications, rather than allowing isolation to be configurable through policy. See "Comparing Endpoint Technologies for Malware Protection" for more information on containment solutions.
Isolation, monitoring and journaling are not common capabilities across all solutions in the application control space, not even in the larger realm of endpoint security solutions. Monitoring endpoints for malicious events is receiving ample exposure in the form of endpoint detection and response (EDR) solutions. Other solutions, such as Webroot, focus more on application control and include ongoing monitoring, isolation and/or change-recording of untrusted applications. Some solutions, such as Symantec’s Data Center Security solutions, include application restrictions for all server applications. Kaspersky Lab includes application restrictions, as well as application whitelisting, as part of its EPP solution.

User Privileges Versus Application Privileges

Application control solutions focus on restricting the execution of applications. Yet, an application’s rights are tied to the privileges of the user executing it. Most attacks require elevated privileges to do major harm and to hide well. Removing privileges from users makes these deeds harder. For example, ransomware executing with user privileges can encrypt all user files. By contrast, ransomware executing with administrative privileges can not only encrypt all user files, but also evade detection by EPP solutions and overwrite shadow copies of files.

The vast majority of enterprise users do not need administrative privileges on modern OSs. In tightly managed environments, software installation is a central process, and most, if not all, users do not require privileges. In lightly managed environments, where central deployment is lacking or many users are mobile, the use of privileges must be controlled. The most lightweight solution for Windows is the native User Account Control (UAC), which lacks reporting and granular policy and exception management.
Issues arise when organizations have applications — or even a single application — that require the user to have local administrative rights. Large organizations frequently have some applications (often legacy) that function only when the user is acting as a local administrator. Providing privileges to all users for such applications is overkill.

Remove local administrative rights from all but the most dynamic endpoints. If required, leverage privilege management solutions to give users privileged access to incidental applications that can't run without it, and to specific system services (such as changing date and time, defragmenting, adding hardware, and enabling security functionality). For the most dynamic endpoints, allow users to self-elevate.

Third-party privilege management solutions, such as Avecto, BeyondTrust, CyberArk, Landesk and Thycotic, greatly extend UAC's capabilities by offering the following:

- Discovery of applications that require privileges
- Central policy and exception management of applications that require privileges
- Central monitoring, including logging of user-provided business justifications

Most Windows privilege management solutions include capabilities that are similar to those of application control solutions. For example, privilege management solutions are often capable of running an inventory to discover applications that require administrative privileges. They also support exception management processes and dynamic policies to preauthorize, as well as automate, privilege elevation. With all these capabilities, it should not come as a surprise that many privilege management solutions include application control functionality.

Application Control Solutions

Categorizing each vendor by its broader focus, Figure 6 provides a sample of vendors in the application control space. Note that various vendors are active in multiple areas, so the categorization is for illustrative purposes only. Although Figure 6 may suggest a very crowded market for application control solutions, this is not the case. The differentiation between products is often large. The only thing some of these products have in common is the fact that they include application control capabilities.

Consider the broader vendor focus in Figure 6 when selecting an application control solution. For example, consider one of the vendors with a broader focus on privileged access management if (1) your main objective is reducing the risk of malware by removing user privileges on client endpoints, and (2) your secondary use case is application control. If your objective is to enhance endpoint protection with a separate or integrated application control agent, consider the vendors with a broader focus on advanced threat protection and endpoint protection. If you want a solution that is
part of, or that tightly integrates with, your existing system management solution, consider vendors
with a broader focus on system management first.

The most important differentiators among the solutions are:

- **Default-deny whitelisting support.** Some solutions lack centralized inventories across all
  applications and endpoints. Others lack a rich set of trusted changers to support dynamic
  endpoints.

- **Privilege management support.** Most solutions focus on application control and do not offer
  the option to manage Windows, or Mac OS X, privileges. Likewise, privilege management
  solutions may include application control functionality, but they often lack capabilities for
  default-deny whitelisting implementations.

- **Postexecution application control.** Only a few solutions offer postexecution isolation,
  monitoring and journaling.

- **Reporting.** Most solutions include consoles to create custom reports. For more comprehensive
  reporting, look for a vendor that enables you to include additional data and that provides
  visibility beyond the product itself.

- **Trusted sources and changers.** To support application whitelisting on client endpoints,
  solutions may include trusted sources and trusted changers. The breadth and usability of these
  items can vary considerably.

- **Integration.** Solutions differ in their ability to integrate with service desk solutions (for managing
  exceptions), system management solutions (for managing policies, generating reports, and
  providing endpoint and application context), and security solutions (for providing security
  context and sharing threat intelligence).

- **Nonapplication control capabilities.** To compensate for gaps in application control, vendors
  may offer additional security capabilities on the endpoint. These include memory protection and
  exploit prevention, vulnerability management, configuration management/control, and EDR.
  Support for these capabilities varies widely among the vendors.

- **Native threat intelligence.** Most solutions integrate with at least some threat intelligence
  providers to render trust ratings for files. However, there is great variation between what is
  included and what "could potentially be integrated."

- **Expertise.** Because of their close relationships with the most popular software vendors, some
  application control vendors are more adept than others at identifying good and bad files. Such
  expertise reduces FPRs.
Various acquisitions took place recently. CyberArk acquired Viewfinity, Thycotic acquired Arellia, Digital Guardian acquired Savant Protection, Heat Software acquired Lumension, and Landesk acquired AppSense. The privileged access management space was especially in flux, with various privilege management solutions finding new homes with the larger privileged access management vendors. With privilege management, whitelisting and content isolation, Defendpoint moved Avecto into the endpoint protection space.

The traditionally strong application whitelisting offerings (Carbon Black, Intel Security and Heat Software) are no longer the only choices for default-deny application whitelisting. However, they are still considered very strong solutions for this use case.

EPP vendors are expanding their application control capabilities. Intel Security was one of the earlier EPP vendors to offer an application whitelisting solution. Intel Security now integrates application whitelisting with its McAfee Threat Intelligence Exchange (TIE) offering for local reputation intelligence. Symantec includes application containment and whitelisting as part of its Data Center Security solutions, but not for client endpoints. Trend Micro includes application control in its EPP suite and also offers application control as a stand-alone solution. In 3Q16, Trend Micro plans to support application control for cloud/server workloads in its Deep Security offering. Kaspersky Lab has fully integrated application control capabilities into its EPP solution. It does not offer these capabilities as a stand-alone solution.
Both Panda Security and Webroot control applications by attesting the security of any unknown applications before allowing them to run.

For more information on the capabilities of the sample vendors mentioned in this section, see The Details section. (Note that only the vendors with application whitelisting and privilege management capabilities are included.)

The Impact of Application Control on Users and IT Operations

If application control is such a strong anti-malware solution for endpoints, why doesn’t everyone use it? Those who have deployed it know the answer: Rolling out application control is not a panacea. Unless they want to end up in a superficial monitoring-only scenario, all application control deployments will have an impact on users and on IT operations.

This section discusses the impact of application control on users and operations, and includes best practices to limit such impact.

Limiting the Impact on Users

Users will like the low CPU and memory overhead of application control solutions, but they may hate the loss of freedom resulting from strict default-deny policies. Organizations deploying application whitelisting often mention web-conferencing platforms as an example.

Web-conferencing solutions are used very frequently, and they often require privileged access during installation. Dozens of such web-conferencing solutions exist, often receiving automatic updates over the internet, sometimes signed using new certificates. If you are considering application control in your organization, analyze this as an example use case: How will your selected solution, your policies and your exception management process support users that want to dial in to a web conference starting minutes from now, using an application not yet covered by your current whitelist?

Organizations must explain the reasoning behind application control to their users. With ransomware attacks affecting organizations across the globe in a very visible way, application control should not be as hard of a sell as it once was (when chief information security officers [CISOs] had to gain business support to protect against "advanced persistent threats").

User awareness aside, organizations must seriously consider the user impact before deploying application control solutions. The application control capabilities that have the most positive effect on user acceptability are dynamic policies and efficient exception management.

For client endpoints, use the following best practices to limit user impact:

- **Partner with your users** to understand why they perform actions in a certain way. Understand their use cases, and create user groupings based on business roles or functions, not on the corporate reporting structure.
- **Predict and automate** most exceptions by using trusted changers. All changes made through trusted changers are fully transparent to users.

- **Staff and educate** IT operations for a quick turnaround in exception management. Make users aware of how to request exceptions and of what turnarounds to expect.

- **Consider self-management** of exceptions and privilege elevation, but only for the most dynamic endpoints. Set time aside to review and potentially correct such preauthorized exceptions.

A common lesson learned from successful application control deployments is to **prepare**. Successful deployments are all about setting the right policies and optimizing the exception management workflow. For more information on what policies work, see the Making Application Control Work section.

### Limiting the Impact on IT Operations

At a high level, the steps for deploying application control are:

1. **Categorize your endpoints** based on (1) the level of security you want application control to enforce, and (2) the flexibility/dynamics you require. Most organizations end up with between four and 10 endpoint categories. Examples of categories include programmable logic controller/supervisory control and data acquisition (PLC/SCADA) devices, point-of-sale systems, critical servers, generic servers, generic endpoints, and dynamic endpoints.

2. **Create a common operating system build.** Apply all outstanding security patches, reapply global policies, and upgrade software. Doing so will remediate misconfigurations and out-of-date configurations, reducing the number of possible exceptions to your application control baseline.

3. **Run inventories** of applications for each category of endpoints.

4. **Build baselines** of trusted applications per endpoint category.

5. **Define trusted changers.** Start with adding your system management system as a trusted changer. Add other trusted changers, such as file locations and trusted administrators, for specific use cases.

6. **Monitor.** Run policies in monitor-only mode for at least two to three weeks. Adjust policies during the process.

7. **Enforce.** As a last step, flip the switch from monitoring to enforcement. You can use the event-monitoring data to gain insight into application usage risks. Even though enforcement may not be necessary in every situation, it should be the end goal for application control.

Due to the complexity of application control solutions and the direct impact of ill-aligned policies and exception management processes on user productivity, the initial deployment of application control can be resource-intensive and often requires a team of specialists. Organizations that are new to application control should strongly consider engaging professional services from the vendor.
or its partners during the deployment phase. Professional service experts can document and transfer their expertise regarding policy and exception management to the operations team.

After the rollout, the application control solution is managed by IT or security operations. Similar to the last section, where we discussed limiting the impact on end users, ongoing success is all about exception management and monitoring. The following best practices will limit the impact on operations during the run phase of application control:

- **Choose trusted changers** to automate as many predictable changes to applications on endpoints as possible.
- **Integrate with system management suites** to simplify and streamline exception management.
- **Selectively employ self-authorization** with business justification for the most dynamic endpoints. Plan for monitoring if you allow self-authorization.
- **Provide information on file context** to IT operations teams so that they can make informed decisions during exception management. Look for solutions that integrate with VirusTotal, network sandboxes, vulnerability assessment solutions and other security technologies.
- **Develop an escalation path** with your vendor, and understand how to react during operational issues. Put the appropriate documentation in place for your end users and your IT operations desk.

**Making Application Control Work**

Figure 7 illustrates the effect that smart policies, efficient exception management and optimal tools will have on operational overhead. Most organizations with successful application control deployments estimate that their operational overhead is less than a full-time equivalent (FTE). Some organizations spend only a few hours a week evaluating exceptions. Others, with more restrictive policies, report that their operational overhead is comparable to the effort of managing an enterprise EPP solution.

**Figure 7. The Success of an Application Control Deployment Depends on Smart Policies and Tools**

Source: Gartner (July 2016)

Figure 1, presented at the beginning of this assessment, summarizes the best practices for limiting the user impact and operational overhead of application control. The following provides a deeper explanation:
For critical servers, fixed-function devices and manufacturing environments, choose default-deny whitelisting (consider per-device whitelists for the most critical endpoints). Control changes very tightly for such endpoints.

For other servers and standard endpoints, choose default-deny whitelisting, but use both a carefully designed policy with trusted changers and an efficient exception management process.

For the most dynamic endpoints, such as developer workstations and IT department workstations, consider self-authorization, monitoring and blacklisting, rather than default-deny whitelisting. Default-deny whitelisting may result in too many exceptions or in too high of a user impact. To limit the impact of malware that gets onto these endpoints, implement compensating controls, such as network zoning and a secure web gateway from a leading vendor.

For endpoints that require both strong security and great flexibility, do not consider application whitelisting as the main control of choice. Rather, look for containment solutions, alternative solutions or compensating solutions. Alternatively, reduce the flexibility requirements (but keep the security requirements), or accept more risk (and keep the flexibility requirements).

Future Developments

There is hardly a reason for any buyer to want a separate application control solution, with a separate management console and separate endpoint agents. Therefore, existing solutions will evolve to become more complete. They will include more application control functionality (for example, isolation) and more adjacent endpoint security technologies (for example, memory control and EDR). The advancement of the market will likely continue along three categories of solutions: endpoint security, system management and privileged access management.

Solutions will continue to cover more platforms. Apart from the obvious Windows OSs, support for legacy OSs, such as Windows Server 2003 and Windows XP, is already common. Mac OS X continues to expand in enterprises, and solutions have started to support this OS as well.

In theory, application control solutions for client endpoints should also support server platforms, but the reality is different. To be broadly used in data centers or in the cloud, application control solutions must support server platforms (including Linux and Unix flavors) across physical, virtual and cloud environments, and they must include strong file integrity monitoring and control capabilities. Server application control solutions will evolve to support short-lived virtual machines as well as hybrid clouds. They will eventually integrate with flexible application deployment strategies.

EPP vendors will continue to improve their application control capabilities. Some EPP vendors don’t have any application control capabilities, others have separate solutions, but only very few have application control integrated into their EPP suites. Many EPP vendors must take important steps to provide this critical capability to their customers in an integrated fashion.

The separation between user-focused privilege management solutions and application-centric application control solutions will continue to exist. Privilege management will gradually be subsumed by larger suites of enterprise privileged account management solutions. By contrast,
application control solutions will eventually find their homes in larger EPP suites, in endpoint solutions focused on advanced threat protection, and in system management solutions.

**Strengths**

Security architects evaluating application control should consider the following strengths:

- Application whitelisting is a strong endpoint security control, especially for endpoints that can accept stricter default-deny policies.
- Privilege management solutions help by removing local administrative rights from all users.
- Application control has low overhead in terms of CPU and memory. It also runs on a diverse set of current and legacy client, server and fixed-function device platforms.
- By permitting dynamic policy controls, leading solutions make application control feasible with little operational effort. Dynamic policies enable acceptable user impact across all endpoints.

**Weaknesses**

Security architects evaluating application control should consider the following weaknesses:

- Not all malware comes as an executable. Attacks may leverage scripts, configuration files and interpreted code. They can even be the data processed by an application. Some applications, in and of themselves, are platforms that execute potentially malicious code. These include browser plug-ins, Java and PowerShell.
- The inventory of applications may already include malicious files.
- Trusted changers may be misused. Attackers may leverage signed malware (by abusing stolen private keys from trusted certificates, for example), trusted directories, administrative accounts, or any other trusted change mechanism used by the deployment.
- Malware may exploit vulnerable trusted code. Organizations should complement application control with vulnerability management, memory protection, containment and, optionally, EDR.
- Process and policy issues seriously impact the effectiveness of application control. Users and IT operations teams may make mistakes managing policies and exceptions, especially in situations where insufficient context and threat information is provided at the time of decision making.

**Guidance**

Security architects looking to deploy application control solutions should observe the following recommendations:

- **Remove local administrative rights from almost all users.** Technology can help with policy-based exception management.
- **Handle gray files according to the security objective for a particular endpoint.** Choose one of the following options (listed in order from most secure to least secure):
  - Do not allow
  - Attest before allow (via the vendor, a network sandbox or IT operations) and/or isolate
  - Rely on user
  - Monitor only

- **Define policies per endpoint category.** To make application control work, match policies to the flexibility and security requirements for each category. In practice, four to 10 endpoint categories are common. At the very least, you should define the following categories and corresponding policies:
  - Fixed-function and critical server endpoints — Employ default-deny whitelisting with no trusted changers, or with very few trusted changers that are strictly controlled.
  - Standard client and server endpoints — Employ default-deny whitelisting with trusted changers.
  - Dynamic client endpoints — Employ blacklisting, monitoring and/or self-authorization.

- **Limit user impact.** For client endpoints, use the following best practices to limit user impact:
  - Predict and automate most exceptions by using trusted changers. All changes made through trusted changers are fully transparent to users.
  - Staff and educate IT operations for a quick turnaround in exception management. Make users aware of how to request exceptions and of what turnarounds to expect.

- **Keep application control manageable.** The following best practices will limit the impact on operations during the run phase of application control:
  - Choose trusted changers to automate as many predictable changes to applications on endpoints as possible.
  - Integrate with system management suites to simplify and streamline exception management.
  - Selectively employ self-authorization with business justification for the most dynamic endpoints. Plan for monitoring if you allow self-authorization.
  - Provide information on file context to IT operations teams so that they can make informed decisions during exception management. Look for solutions that integrate with VirusTotal, network sandboxes, vulnerability assessment solutions and other security technologies.

- **Complement application control with technologies that:**
  - Protect against pure in-memory attacks, if this is not adequately covered by the application control solution itself. If you have best practices for vulnerability management in place,
consider exploit technique mitigation solutions, such as Microsoft EMET and Palo Alto Networks Traps, to protect against exploitation of whitelisted applications.

- Protect against data-driven attacks. Many whitelisted applications are capable of executing code. Examples include web browsers and add-ons, Microsoft Office macros, and Windows Script Host.
- Protect against configuration attacks.
- Compensate for lenient policies (monitoring, blacklisting and self-authorization) on dynamic endpoints by leveraging additional security controls, such as containment, EDR or behavior-based malware protection.

**Choose a solution that best fits your use case.** No single solution is best-of-breed across clients, servers and fixed-function devices. Moreover, no single solution supports all privilege management, default-deny whitelisting and application isolation capabilities. Thus, you must determine which solution works best for your requirements.

**The Details**

Table 1 provides details on solutions mentioned in the Application Control Solutions section. Only the vendors that have application whitelisting and privilege management capabilities are included.
Table 1. Capabilities for Select Application Control Solutions

<table>
<thead>
<tr>
<th>Solution</th>
<th>Primary Use Cases</th>
<th>Secondary Use Cases</th>
<th>Broader Vendor Focus</th>
<th>Files in Scope¹</th>
<th>Application Reputation Sources (Internal and Third-Party)</th>
<th>Technologies to Handle Unknown Files</th>
<th>Additional Capabilities to Mitigate Application Control Limitations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avecto DefendPoint</td>
<td>Windows and Mac OS X privilege management Application control</td>
<td>-</td>
<td>Endpoint protection</td>
<td>PE files, installers, Windows scripts, PowerShell, batch files, registry, ActiveX, Microsoft Management Console, COM objects and OS features</td>
<td>McAfee TIE VirusTotal</td>
<td>Default-deny policies, isolation or exception management Containment of applications that process untrusted content, such as Word and PDF documents</td>
<td>Combination of least privilege, application whitelisting and isolation to protect against malicious content that can exploit known good applications</td>
</tr>
<tr>
<td>BeyondTrust PowerBroker</td>
<td>Windows and Mac OS X privilege management Dynamic risk-based application control</td>
<td>Session monitoring File integrity monitoring</td>
<td>Privileged access management</td>
<td>PE files, Windows scripts, installers, Windows Update, ActiveX, third-party management consoles, PowerShell, batch files, COM objects and OS features</td>
<td>Patented Vulnerability-Based Application Management (based on Retina CS and VirusTotal)</td>
<td>Policy-based approaches to: Reduce privileges Deny applications Quarantine applications Implement patch management</td>
<td>Event log monitoring File integrity monitoring Keystroke logging Session recording User-based justifications Automated quarantine Continuous vulnerability assessment Extensive third-party integration and patents for password-safe technology</td>
</tr>
<tr>
<td>Solution</td>
<td>Primary Use Cases</td>
<td>Secondary Use Cases</td>
<td>Broader Vendor Focus</td>
<td>Files in Scope¹</td>
<td>Application Reputation Sources (Internal and Third-Party)</td>
<td>Technologies to Handle Unknown Files</td>
<td>Additional Capabilities to Mitigate Application Control Limitations</td>
</tr>
<tr>
<td>----------</td>
<td>------------------</td>
<td>---------------------</td>
<td>----------------------</td>
<td>-----------------</td>
<td>----------------------------------------------------------</td>
<td>-------------------------------------</td>
<td>---------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Carbon Black Enterprise Protection</strong></td>
<td>Application control</td>
<td>Malware detection and prevention</td>
<td>Endpoint protection</td>
<td>Any files, including PE files, installers, batch files, Windows scripts, PowerShell and ActiveX</td>
<td>Carbon Black Threat Intel, which includes native and third-party intelligence services</td>
<td>Default-deny and detonate-and-deny modes</td>
<td>Integration with sandboxes, including Palo Alto Networks Wildfire, Lastline, Check Point and others, via an open API Advanced threat indicators to deal with the unknown Trust-driven policies</td>
</tr>
<tr>
<td><strong>CyberArk Viewfinity</strong></td>
<td>Privilege management</td>
<td>Application control/whitelisting</td>
<td>Privileged account security</td>
<td>PE files, installers, Microsoft Update, COM objects, Windows scripts, PowerShell, batch files, registry and ActiveX</td>
<td>Viewfinity file reputation algorithm VirusTotal NSRL Check Point Palo Alto Networks FireEye</td>
<td>Default-deny whitelisting Isolation and restricted mode (e.g., restrict access to network, local disk, registry, network share or process memory)</td>
<td>Process memory protection Restricted mode (to balance security, usability and manageability) Integration with third-party solutions</td>
</tr>
<tr>
<td><strong>Digital Guardian Application Whitelisting (DGAW)</strong></td>
<td>Application whitelisting</td>
<td>-</td>
<td>Endpoint protection</td>
<td>PE files, installers and interpreters (As specified by the DGAW administrator; examples include, but are not limited to .py, .php, .cmd and .exe)</td>
<td>VirusTotal</td>
<td>Lockdown mode: New PE files cannot be written to disk; default-deny policies for</td>
<td>Comprehensive endpoint protection via DGAW combined with the Digital Guardian Agent</td>
</tr>
<tr>
<td>Solution</td>
<td>Primary Use Cases</td>
<td>Secondary Use Cases</td>
<td>Broader Vendor Focus</td>
<td>Files in Scope¹</td>
<td>Application Reputation Sources (Internal and Third-Party)</td>
<td>Technologies to Handle Unknown Files</td>
<td>Additional Capabilities to Mitigate Application Control Limitations</td>
</tr>
<tr>
<td>---------------------------------------------------</td>
<td>------------------------------------</td>
<td>-------------------------------------------------</td>
<td>--------------------------------------------------------------------------------------</td>
<td>----------------------------------------------</td>
<td>-----------------------------------------------------------</td>
<td>-------------------------------------</td>
<td>---------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Heat Software Endpoint Management and Security Suite (EMSS)</strong></td>
<td>Application whitelisting</td>
<td>Memory protection</td>
<td>Endpoint protection</td>
<td>PE files and installers</td>
<td>Endpoint Integrity Service</td>
<td>Default-deny whitelisting plus Trust Engine, which includes:</td>
<td>Memory protection to detect and block memory injection attacks, which would otherwise bypass application control</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EPP (including patches and remediation, antivirus, and device control along with application control)</td>
<td>Unified endpoint management and cloud service management</td>
<td></td>
<td></td>
<td>Trusted Updater</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Trusted Publisher</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Trusted Path</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Local Authorization</td>
<td></td>
</tr>
<tr>
<td><strong>Intel Security McAfee Application Control</strong></td>
<td>Application whitelisting with default-deny, detect-and-deny, and verify-and-deny options</td>
<td>Inventory monitoring</td>
<td>Server, endpoint, web and network protection across physical, virtual and cloud deployments</td>
<td>PE files, installers, Windows scripts, PowerShell, batch files, ActiveX, interpreters (Java, Perl, Python and others), COM objects and Linux executables</td>
<td>McAfee GTI McAfee TIE Integration with third-party feeds</td>
<td>Default-deny mode (approved whitelist and trusted updaters)</td>
<td>Dynamic trust model</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Change monitoring and control for nonexecutable files</td>
<td></td>
<td></td>
<td>McAfee GTI Integration with third-party feeds</td>
<td>Detect-and-deny mode (file and certificate reputation)</td>
<td>User notifications</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Verify-and-deny mode (sandbox testing)</td>
<td>User self-approval</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>McAfee ePO (integration with other EPP and EDR capabilities)</td>
</tr>
<tr>
<td>Solution</td>
<td>Primary Use Cases</td>
<td>Secondary Use Cases</td>
<td>Broader Vendor Focus</td>
<td>Files in Scope¹</td>
<td>Application Reputation Sources (Internal and Third-Party)</td>
<td>Technologies to Handle Unknown Files</td>
<td>Additional Capabilities to Mitigate Application Control Limitations</td>
</tr>
<tr>
<td>---------------------------</td>
<td>--------------------------------------------------------</td>
<td>---------------------</td>
<td>----------------------</td>
<td>------------------</td>
<td>-------------------------------------------------------------</td>
<td>--------------------------------------</td>
<td>---------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Kaspersky Lab Application Control</strong></td>
<td>Application whitelisting Default-deny mode Postexecution application behavior control Application privilege control (isolation)</td>
<td>Vulnerability assessment Patch management Advanced exploit prevention</td>
<td>Endpoint protection, including security for data center and virtual environments System management, including mobile device security and management Advanced threat protection Web and email attack vector protection</td>
<td>PE files, installers, Windows scripts, PowerShell, batch files, registry, Microsoft Management Console, interpreters (Java), HTML Application and Windows Runtime applications (C# and JavaScript-based)</td>
<td>Kaspersky Security Network Certificates for reputation from the most popular software vendors Heuristics and mathematical algorithms to classify software as good or bad</td>
<td>Postexecution application behavior control Application privilege control (isolation) Vulnerability and patch management Advanced exploit prevention Anti-malware</td>
<td>Special handling of installers Signatureless memory protection Buffer overflow protection ROP mitigations</td>
</tr>
<tr>
<td><strong>Landesk Security Suite and</strong></td>
<td>Application whitelisting Contextual control over authorized</td>
<td>Workspace management: Security</td>
<td>PE files, installers, Windows scripts, ActiveX, registry, batch files and Windows Store applications (UWP)</td>
<td></td>
<td>AppSense: None</td>
<td>Whitelist (with a focus on low ad-</td>
<td>Granular system management, patch manage-</td>
</tr>
<tr>
<td>Solution</td>
<td>Primary Use Cases</td>
<td>Secondary Use Cases</td>
<td>Broader Vendor Focus</td>
<td>Files in Scope¹</td>
<td>Application Reputation Sources (Internal and Third-Party)</td>
<td>Technologies to Handle Unknown Files</td>
<td>Additional Capabilities to Mitigate Application Control Limitations</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>------------------------------------------</td>
<td>---------------------------------------</td>
<td>----------------------</td>
<td>------------------</td>
<td>------------------------------------------------------------</td>
<td>--------------------------------------</td>
<td>---------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>AppSense Application Manager</strong></td>
<td>and access control</td>
<td>applications for compliance and licensing URL-based control for web-based applications and network access control</td>
<td>cure, configure, optimize, and personalize both physical and virtual Windows desktops Patch management Asset management Service management System management</td>
<td>Landesk: Kaspersky Lab file reputation database (OEM) to help identify the reputation of discovered files</td>
<td>ministrative overhead) Self-authorization Autotrust mode: Run if reputation is good Learning mode Integration with Bufferzone for isolation, if desired</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Microsoft AppLocker</strong></td>
<td>Application whitelisting</td>
<td>-</td>
<td>Software, electronics PC hardware Services</td>
<td>PE files, installers, Windows scripts, PowerShell, batch files, ActiveX and packaged apps (.appx)</td>
<td>-</td>
<td>Default-deny whitelisting</td>
<td></td>
</tr>
<tr>
<td><strong>Symantec Data Center Security</strong></td>
<td>Application whitelisting and isolation</td>
<td>Security event and real-time file monitoring</td>
<td>Data center protection and monitoring</td>
<td>PE files, installers, Windows scripts, PowerShell, batch files, Microsoft Update, ActiveX, HTML Application, program information file, screensaver, Mi-</td>
<td>Symantec Insight</td>
<td>Monitor all activity Isolate Deny</td>
<td></td>
</tr>
</tbody>
</table>

¹ Files in Scope refers to the types of files that are considered when evaluating application Reputation Sources (Internal and Third-Party) and technologies to handle unknown files. These files include PE files, installers, Windows scripts, PowerShell, batch files, Microsoft Update, ActiveX, HTML Application, program information file, screensaver, etc.
<table>
<thead>
<tr>
<th>Solution</th>
<th>Primary Use Cases</th>
<th>Secondary Use Cases</th>
<th>Broader Vendor Focus</th>
<th>Files in Scope¹</th>
<th>Application Reputation Sources (Internal and Third-Party)</th>
<th>Technologies to Handle Unknown Files</th>
<th>Additional Capabilities to Mitigate Application Control Limitations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thycotic Application Control Solution</td>
<td>Application whitelisting Application blacklisting Application graylisting</td>
<td>Privilege management Least privilege</td>
<td>Privileged account management</td>
<td>PE files, installers, .mst, Windows scripts, PowerShell, batch files and ActiveX</td>
<td>VirusTotal Kaspersky Whitelist Symantec Landesk Microsoft SCCM Reference systems Software catalogs Customer-defined reputation sources</td>
<td>Graylisting (situational awareness) Process isolation Application firewalling Real-time elevation approval request Application sandboxing (integration with Symantec Workspace Virtualization)</td>
<td>Application-level firewall Memory protection Integration with Microsoft exploit mitigation (EMET) Real-time elevation approval request SIEM integration</td>
</tr>
<tr>
<td>Trend Micro Endpoint Application Control</td>
<td>Application whitelisting</td>
<td>Application monitoring and journaling</td>
<td>Endpoint, gateway, network and data center security</td>
<td>PE files and Windows Runtime applications (C++, .NET and HTML5 with JavaScript)</td>
<td>Trend Micro Smart Protection Network, including: Certified Safe Software (known good) Air Score (current vulnerabilities and patching frequency)</td>
<td>System lockdown Default-deny policies Selective application blacklisting Selective application whitelisting Trusted sources of change End-user approval requests</td>
<td>Central management of all endpoint modules (Control Manager) Remote code exploitation (Vulnerability Protection) System change and forensic analysis (Endpoint Sensor) Memory protection and behavio-</td>
</tr>
<tr>
<td>Solution</td>
<td>Primary Use Cases</td>
<td>Secondary Use Cases</td>
<td>Broader Vendor Focus</td>
<td>Files in Scope¹</td>
<td>Application Reputation Sources (Internal and Third-Party)</td>
<td>Technologies to Handle Unknown Files</td>
<td>Additional Capabilities to Mitigate Application Control Limitations</td>
</tr>
<tr>
<td>----------</td>
<td>------------------</td>
<td>---------------------</td>
<td>----------------------</td>
<td>-----------------</td>
<td>----------------------------------------------------------</td>
<td>-------------------------------------</td>
<td>---------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Regional and global usage</td>
<td>ral analysis (OfficeScan)</td>
<td></td>
</tr>
</tbody>
</table>

ePO = ePolicy Orchestrator; GTI = Global Threat Intelligence; ROP = return-oriented programming; SCCM = System Center Configuration Manager; SCEP = System Center Endpoint Protection; SIEM = security information and event management; TIE = Threat Intelligence Exchange; UWP = Universal Windows Platform.

¹ The following file types are associated with the categories:
Portable Executable (PE) files: .acm, .ax, .cpl, .dll, .drv, .efi, .exe, .mui, .ocx, .scr, .sys and .tsp. Support for file types varies; the most typical are .dll, .exe, .scr and .ocx.
Installers: .msi and .msp.
Windows scripts: .wsf, JScript (.js and .jse) and VBScript (.vbs and .vbse).
PowerShell: .ps1.
Batch files: .bat and .cmd.
Registry: .reg.
Microsoft Management Console: .msc.
HTML Application: .hta.
Program information file: .pif.

Source: Gartner (July 2016)
Gartner Recommended Reading

Some documents may not be available as part of your current Gartner subscription.

"Comparing Endpoint Technologies for Malware Protection"

"Evaluation Criteria for Endpoint Protection Platforms"

"How to Successfully Deploy Application Control"

"Market Guide for Privileged Access Management"