Virsec Deterministic Protection Platform

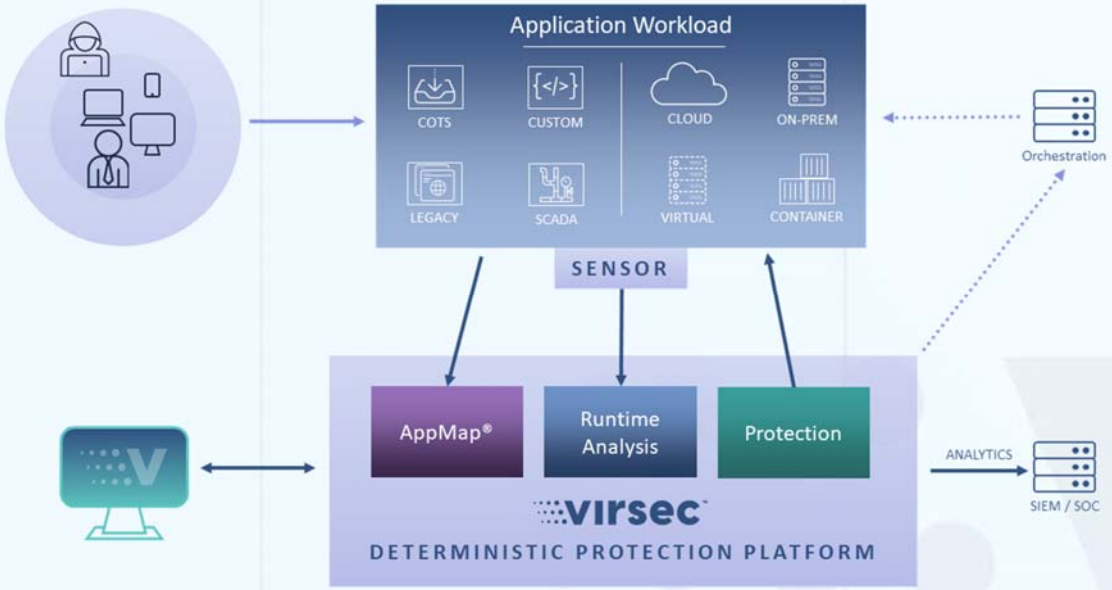
How it works

Virsec Deterministic Protection Platform

The Virsec Deterministic Protection Platform (DPP) leverages the patented AppMap™ technology to protect high-value enterprise applications from highly sophisticated attacks, including memory corruption, code injection, supply chain poisoning, and several others. DPP ensures that applications only execute as intended and restrains bad actors from hijacking control of the application and subsequently stealing or destroying high-value enterprise data.

Conventional cybersecurity tools are designed to extract markers from user, device, network, process, and file system activities observed on the workload. These runtime marker sequences are compared to known and cataloged attack techniques. When attacker techniques vary beyond what has been previously cataloged, their attacks will bypass conventional security tools and will succeed. For most enterprises, it is a massive logistical challenge to constantly keep updating these heuristic and behavioral marker sequences of their deployed security solutions and to keep patching their applications as soon as possible to reduce the risk of being attacked.

DPP eliminates the logistical nightmare of keeping up with vulnerabilities and does not require the ongoing update of threat feeds. DPP monitors applications instead of attack techniques and prevents "foreign" code from running on the workload. DPP's technique of creating application guardrails ensures that protected applications execute only the code the developer intended. This approach relieves enterprises from the detect-and-respond model, as even unpatched vulnerable applications protected by DPP, cannot be exploited.



Reduce Attack Surface, Enforce Trusted Execution

Virsec DPP uses strict, yet easy-to-deploy, application control policies to reduce the attack surface and enforce a zero-trust model that ensures only safe and trusted software is allowed to run on workloads.

Process and Library Monitoring

Virsec supports advanced allow-listing techniques to ensure that the integrity of workloads is not compromised. This is achieved by creating a checksum-based allow-list from either baselining individual hosts or scanning a reference host, thus creating a chain of trust that can be traced to the code provider. Executables are automatically added to the allow-list based on various trust factors, including but not limited to OS validations, publisher certificate, and file reputation (via integrations with Reversing Labs and VirusTotal).

Zero Dwell Time

Virsec stops execution of malicious processes and library injection/hijacking in real-time before they can execute. This contrasts with many detect-and-respond based solutions where malicious actors can do most of the damage before they are detected.

Script and Dynamic Application Control

Virsec stops file-less and living-off-the-land attacks that target trusted applications like shell/script interpreters (PowerShell, wscript, mshta, bash) and system processes like svchost, rundll32, and more. It leverages application control policies that apply granular controls over the dynamic behavior of these processes. It can restrict usage to certain trusted command lines arguments, users, and parent-child process lineage, and stop known attack tactics and techniques.

Runtime Monitoring and Protection

Memory Exploit Protection

Virsec stops attempts to inject and run malicious code from memory by targeting trusted processes. It stops process injection techniques including, but not limited to, Process Hollowing, and Process Doppelgänging. These exploit techniques are detected and stopped in real-time without the need for any signature, learning, or customization whatsoever.

File System Monitoring (FSM)

Virsec monitors application and system-critical folders for malicious changes. FSM generates events for the creation, modification, permission change, and deletion of files in the monitored regions of the file system. Additionally, the feature reports any changes in access privileges and file ownership in the monitored folders. It thwarts tampering of web or application servers that typically remain static but may be attacked through these avenues. Tracked attacks include path traversal vulnerabilities and file upload/download vulnerabilities. It supports explicit inclusion and exclusion of specified file extensions (like .tmp, .log) and folders within the monitored folders of the file system. FSM also aids in detecting and stopping ransomware attacks.

Threat Protection Engine

Virsec plugs into third-party APIs (such as NetApp's SnapLock, and Alliance IT's Sentry Wire appliances) and deploys remedial actions in response to threats/attacks detected by the Virsec Deterministic Protection Platform. Incident Response teams can script customized call-backs into the Protection Engine's configuration file or simply leverage standard out-of-the-box protection actions. These actions are then applied by the Virsec Platform on the impacted workload where an Incident occurred.

The Virsec Protection Engine also communicates granular attack attribution information to third-party systems or scripts. Virsec enriches the threat feeds for those cyber security solutions including AI/ML based solutions that depend on cataloging attackers' newest techniques. The data includes time of attack, library name, process name, and attacker IP address to provide context for the appropriate protective action. These actions are in addition to inline protection capabilities that can stop attacks out-of-the-box.

Advantages of Virsec DPP

Virsec uses a highly deterministic approach that does not rely on probability-based detection models and generates zero false positives. Recognized by 80+ patents, innovation is the backbone of DPP's various techniques to truly protect web-based and binary applications from the inside-out. Rather than focusing on constantly evolving system behaviors, DPP monitors actual code execution, effectively determining if the code is executing as intended by the developer or is being derailed by attackers. To do so, DPP leverages three highly differentiated and deterministic approaches:

Control Flow Integrity (CFI) – protects pre-compiled code used in high-speed web servers whose code is often written in languages like C/C++.

Byte Code Instrumentation (BCI) – protects application servers that run the business-logic of web-based applications.

Application Control Engine (ACE) – allows only known developer-provided executables, libraries, and scripts to run, and blocks execution of unknown file-based or file-less malware.

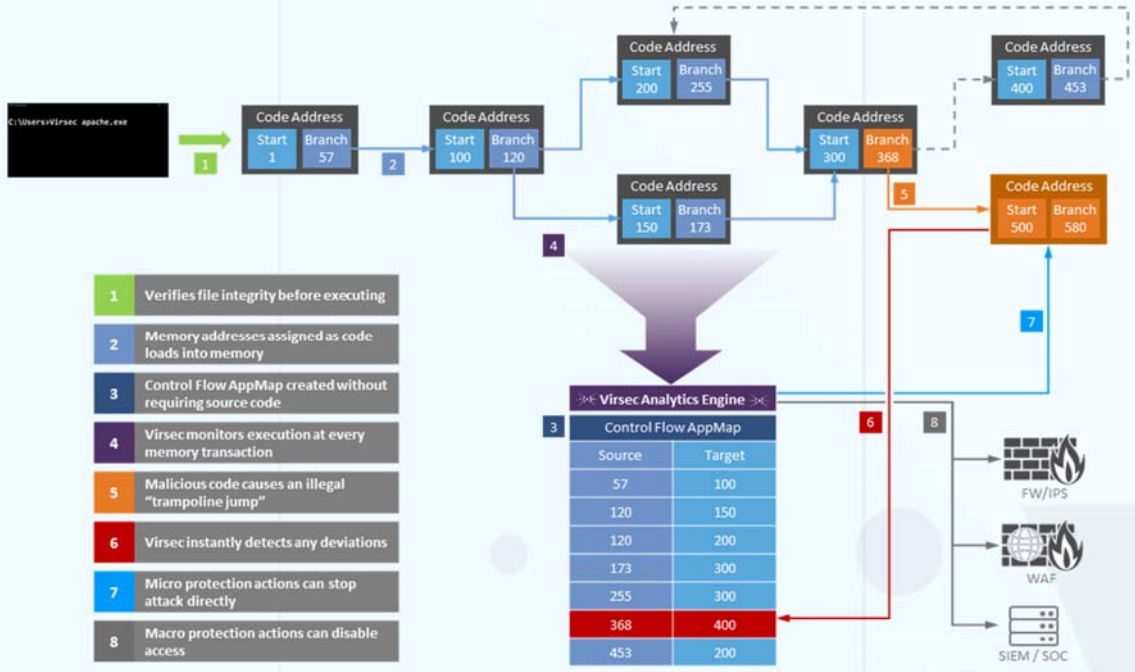
How Control Flow Integrity (CFI) Works

An application's pre-compiled code is a collection of machine instructions grouped into basic blocks. Individual basic blocks execute in a specific order reflecting the intent of the application. This sequential collection of blocks and their traversal in memory at runtime is captured by Virsec and displayed in our unique Control Flow AppMap. Each application has its own individual AppMap, which never changes until the developer releases a new version of the application. CFI monitoring techniques allow the execution of the application and continuously compare actual execution to the Control Flow AppMap. If the execution deviates from the AppMap, this is a definitive indication that the application is no longer executing the developer's intent. A CFI process that granularly tracks execution can defend against highly sophisticated attack techniques including Buffer Overflows, Return Oriented and Jump Oriented Programming gadgets (ROP and JOP), and more.

Virsec's Implementation of CFI

When an application is initially loaded into memory, DPP's CFI technology automatically extracts the AppMap for every executable and shares it with the Virsec Analysis Engine. Virsec's CFI solution also embeds a sensor into each basic block of the application. When the application executes a transition between blocks, the sensor compares the source and origin of that transition to the AppMap. If CFI detects any deviation of the memory boundaries, this is a deterministic sign of attack, and the system can take protective actions within milliseconds. Protective responses can include out-of-band alerts – such as sending the security team a notification, or inline mitigations – such as precisely removing offending code from an application without affecting other legitimate users. These protective responses can be applied uniformly across many applications or applied granularly – reacting to specific characteristics of an attack approach or the targeted application.

Virsec's CFI implementation is highly effective against advanced, never-seen-before attacks launched by the most sophisticated attackers. The approach precisely detects attacks the instant they cause deviations in application machine-code, without relying on broad-stroke system behavior, while ignoring attempts to camouflage such attacks.



How Byte Code Instrumentation (BCI) Works

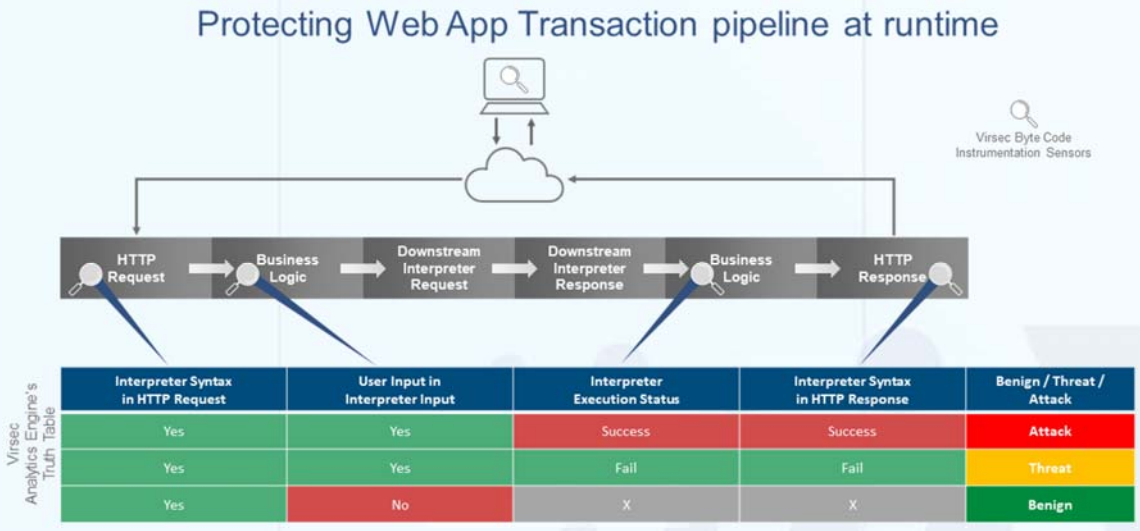
Unlike pre-compiled code, an application's byte code (or interpreted code) is compiled on-demand and is written into a transient section of memory called the native code cache. In the case of interpreters like Java, the native code cache is a fixed size. This means previously compiled code may have to be aged from time to time to accommodate newly compiled methods. Additionally, heavily used byte code may need to be recompiled with even greater optimization.

An application's business logic is constantly generating code for downstream interpreters. Some common interpreters used by web applications include SQL, OS (bash/ DOS etc.), HTTP, XML, JSON, File System, Java, and CLR. Sophisticated attackers often deliver camouflaged payloads consisting of Verbs targeting a downstream interpreter used within the business logic, and then rely on the compiler to turn the payload into code for that downstream interpreter. The inability of the application to perform adequate input sanitization results in attacks on the various downstream interpreters used by the application.

Web protection is enabled using deep web instrumentation in interpreted languages and frameworks like Java, .NET, .NET Core, RoR, PHP, and Node JS that are used globally to write modern web applications. BCI provides Virsec with full visibility into every web request and response, every database query, and the overall application logic and data flow. This visibility, combined with stateful inspection and detection of malicious user inputs, enables Virsec to detect attacks and threats typically categorized under OWASP Top 10, such as SQL and Command Injection, Cross-site Scripting (XSS), CSRF, and many more.

Virsec's Approach to BCI

DPP leverages strategically placed sensors in the customer's application instances using a light touch variable-based approach. Virsec's runtime instrumentation attributes the origin of any generated runtime instructions to the application's own code. For example, a generated SQL statement's Verbs, Predicates, Conditions, and Literals are all attributed to having originated from the application's business logic, while application inputs or Literals are attributed to external users. In addition to detecting attacks in real-time, DPP can also launch protective actions within milliseconds. Since the analysis leverages actual runtime data, Virsec can deliver comprehensive detection with no false positives – unprecedented in the security industry. By comparison, conventional web security solutions like WAFs use learned behavioral techniques, have limited visibility (only HTTP traffic), and often deliver excessive false positives, as do EDRs.



How Application Control Works (ACE)

Application control solutions provide a foundational framework for protecting server workloads from supply chain threats and targeted malware attacks. Unlike personal endpoints where users control the file system and can download arbitrary software, server file systems are more rigid. Operations staff follow well-defined hardening techniques such as vulnerability and change management practices on servers.

An effective application control solution for server workloads must automatically create an allow-list of all essential scripts and executables on the host. This prevents any file or file-less malware, even those never seen before, from executing on the protected host. This model is particularly effective for protecting application servers.

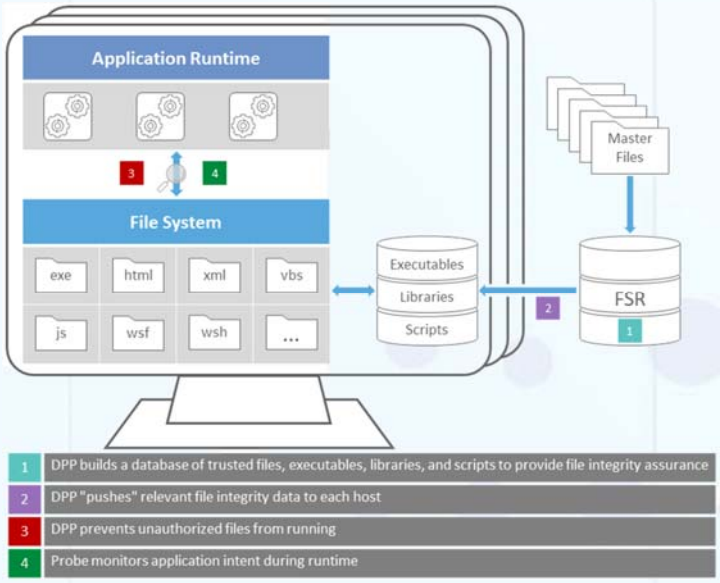
Unfortunately, most existing application control solutions are very difficult to configure and use, as they require a tedious, time-consuming learning approach to identify and freeze executable allow-lists. Any missing but legitimate executable triggers immediate problems for the system admin and disruption to end-users. In addition, any software updates to the OS and application packages cause even more work for the system admin and the user. These operational hassles have frustrated admins and make application control impractical, despite its architectural benefits.

Virsec's Approach to Application Control

DPP unburdens IT by dramatically reducing the effort required to create and maintain allow-lists for server workloads. DPP automatically determines which files, executables, and scripts are part of a server image for all software packages installed on servers. During this process, a package scan captures the name, relative directory path, checksum, and ACL information for each executable and script for that package. This information is consolidated in a centralized System Control Repository.

DPP extends this chain of trust into an AppMap. When a host registers with Virsec and sends details of packages present in the host, DPP responds and provides the checksum and related information for all files in the package. Package upgrades are easily handled in the same way as the original package. Windows OS upgrades are monitored through Microsoft's SFC database which Microsoft upgrades whenever it pushes patches.

This same approach is applied to protecting scripts such as PowerShell. When a new process starts, DPP Process Monitoring inspects the process command line to either allow the application to execute or terminate those processes whose executable is not on the allow-list. In the case of scripts like PowerShell, DPP Process Monitoring examines the authenticity of both the executable and the script file. DPP users experience the full architectural benefits of allow-listing minus the tediousness of legacy approaches.



Conclusion

Compared to the conventional, pervasive detect/respond/remediate model that exists today, the Virsec Deterministic Protection Platform provides a far more effective and practical approach to protecting server workloads and applications. Unlike solutions that rely on heuristics or behavioral rules to detect attacks, DPP uses a deterministic, code-based approach to detect and protect against advanced cyberattacks that cuts attacker dwell time to zero.

Virsec is uniquely able to terminate attacks that cause an application to stop executing its own code and instead start executing code controlled by an attacker. DPP does not generate false positives and can detect attacks that are invisible to conventional security solutions.

Virsec is also unique in providing protection across the full-stack of processes running on a workload whether these are web applications or standalone special purpose binaries. DPP delivers the most comprehensive, accurate, and timely protection against file-based and fileless malware as well as vulnerabilities in pre-compiled and interpreted code.