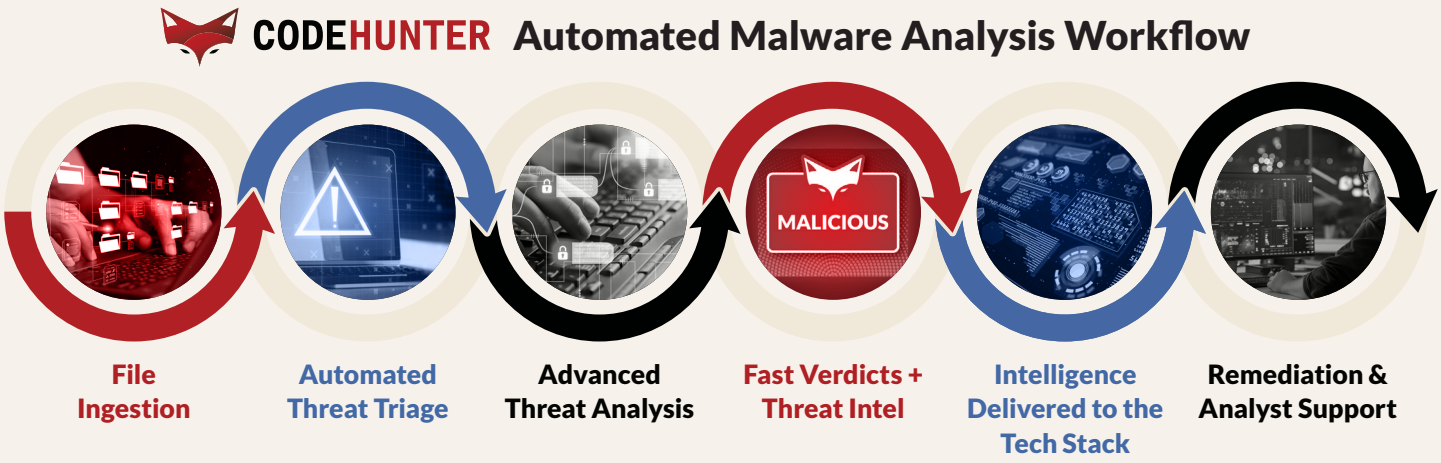
CODEHUNTER

Automated Advanced Malware Analysis Built on Six Patents

CodeHunter leverages patented static, dynamic and AI analysis to identify malware threats traditional defenses miss.



CODEHUNTER AUTOMATES THE ENTIRE MALWARE ANALYSIS PROCESS—from file ingestion to threat remediation—eliminating the need for manual detonation, investigation, and correlation. Files from your security tools or cloud storage are triaged and analyzed using advanced static, dynamic, and AI-driven behavioral techniques. In minutes, you get clear verdicts, threat behavior profiles, and IOCs, all integrated into your SIEM, SOAR, EDR, or TIP.

This patented malware analysis process replaces hours of manual effort, reduces analyst workload, and accelerates detection and response—so your team can act faster and more confidently against advanced threats.

SPECIFIC USE CASE

Our patents enable CodeHunter to automatically recognize complex, ordered behaviors within malware, making analysis more precise and reliable.

CodeHunter's malware analysis engine automatically provides:

- **Detailed threat context** mapped to **MITRE ATT&CK** and the **Malware Behavior Catalog**
- **High-fidelity behavioral insights** that support clear, reliable threat verdicts
- **Actionable intelligence** that accelerates SOC triage and response

"Our newest three patents focus on defining a system to track a sequence of behaviors and corresponding data flow constraints. This lets us identify nuanced behaviors that must occur in a specific order with specific data elements. These enhancements allow our static analysis engine to produce high fidelity behavior matches."

Arion Lawrence
CodeHunter's Chief Technology Officer

PATENT: "Methods and Systems for Identifying Control Flow Patterns and Dataflow Constraints in Software Code"
March 25, 2025
Describes a set of rules written in a machine-readable format to understand the flow of the program and how data moves through it. By analyzing a specific function call in the code, it tracks where the function is called from and what information is passed to it. Then, it evaluates if the function is being used in a specific, expected way that is known to be dangerous or malicious. This BSU language enables detection with high confidence that a behavior exists due to match constraints.

PATENT: "Methods and Systems for Analyzing Dataflow Associated with Software Code to Detect Software Anomalies"
February 11, 2025
The binary code associated with a function call can be indicative of a behavior of interest such as a software anomaly or malware. This binary code can be difficult to interpret, but anomalous behaviors identified can help in understanding the purpose of the code. This patent describes a process to determine the arguments passed when functions are called. In addition, a constrained predicate set is built that can determine if specific argument values are referenced when used with solver software.

PATENT: "Methods and Systems for Identifying Control Flow Patterns in Software Code to Detect Software Anomalies"
January 7, 2025
Analyzing control flow patterns in software code can detect software anomalies, including potential malware. This patent describes a process for extracting a control flow pattern from machine-readable binary code for the purpose of detecting a series of library function calls to detect software anomalies. A signal indicates if the binary code includes one or more library function calls included in the control flow sequence. This innovation refines our existing malware analysis process, enabling accurate and efficient threat behavior identification.

PATENT: "Behavior Specification, Finding Main, and Call Graph Visualizations"
February 5, 2019
This patent describes a behavior specification language used for visualization, whereby a graph can be rendered and color-coded by ancestral relation and function call distance. One of the processes analyzes behavior functionality by processing precise programming behavior abstractions and classifies the code as malicious based on these behaviors. Another calculates the complexity measure given the starting point of execution of a compiled argument. The behavior specification unit (BSU) language currently used in our solution adapted these techniques to explicitly model malware behavior patterns.

PATENT: "Call Trace Generation Via Behavior Computation"
January 1, 2019
This patent describes an approach to verification condition generation that begins by tracking a synthetic call trace, then extending the instruction semantics. Adding a current function call to an existing call trace allows for the extraction of the computed behavior of a program. This computed behavior can then be categorized and analyzed.

PATENT: "Computed Call / Jump Target Resolution Via Behavior Computation"
March 22, 2018
This patent describes a process to prepare a program for behavioral analysis. This method and system figure out where a computer program is trying to go when it makes a jump or call to another part of the program—especially when that destination is calculated on the fly and is not hard-coded. Knowing where the original jump intends to go provides insight into the code's behavior and can help determine if it is demonstrating malicious activity.

Find out how CodeHunter can strengthen your security posture and protect you from unknown threats. ▸ **codehunter.com/learnmore**

CONTACT US TODAY

CODEHUNTER