What Makes HeartSuite Tick? A Technical Overview

HeartSuite takes a radical new approach to security. Instead of trying to detect and isolate malware after the fact like our competition does, we've redesigned the Linux server from the ground up to make it completely secure — by preventing attacks before they have a chance to get off the ground.

Here's what sets HeartSuite apart: Even if malware is downloaded to a HeartSuite server, the architecture of HeartSuite won't allow it to execute its harmful commands, thanks to HeartSuite's unique core features. That means HeartSuite stops malware *before* it can be detected — up to and including zero day attacks.

The Heart of HeartSuite: Core Features

Here's a brief description of the core features that make up HeartSuite's revolutionary approach — and how they work together to bring you unparalleled protection.

❤ Core feature #1: Application Permission Orders

The most essential feature of HeartSuite — and the one that really distinguishes it — is its use of Application Permission Orders (APOs). An APO is a collection of orders that provide access to various files, directories, and network connections. HeartSuite includes programs and scripts for easily creating and modifying APO records as needed.

HeartSuite is comprised of a modified Linux kernel and tools in the form of programs and scripts. The HeartSuite kernel requires a program to have an APO record before it's permitted to run. Once HeartSuite is installed, it will generate messages in both its log and the kernel log whenever a program runs that lacks an APO record. A simple HeartSuite script can be used to automatically scour the logs for this error and then create the needed APO record. Of course, the system administrator can also create such a record manually.

Once the APO record is in place, HeartSuite will generate log messages about the files and directories that the program accesses — and it will also generate log messages about network connections the program makes. HeartSuite's simple script can find these error messages too, and then add appropriate permission orders to the program’s APO record. Likewise, the system administrator can add these orders manually, if preferred.

Access to files and directories is also divided into two categories: read-only and write. That means a file permission order can restrict a program so that it's only allowed to read the file, or it can permit a program to write to a file. (Write permission implies read permission as well.) Permission to access a directory for reading allows a program to not only read the directory, but also to read any file in the directory or its subdirectories. Similarly, write permission permits a program to write to any files in the directory or its subdirectories — including the creation of new files.

The HeartSuite script will also add appropriate permission orders for accessing remote computers. In particular, the IP addresses of computers that a program is trying to access are listed in the log messages. These IP addresses can then be added to the program’s APO record either using the script or manually.

Once all permission orders have been added that are needed by the program, the HeartSuite kernel will no longer generate any error messages about that program’s activities, unless it attempts to access files or network connections that have not yet been permitted. This monitoring doesn't significantly impact performance, either. The HeartSuite caching mechanism loads only a single APO record into memory for a running program — even if there are thousands of concurrent instances of that program — thereby minimizing impact on kernel memory use.

The key is to permit a program to access files and network connections that are necessary for the work that the program needs to do. A Trojan malware program, masquerading as a legitimate program, will need basic file access — but a legitimate program won't need access to all of the data files on the server. So when Trojan malware attempts to access other files —to encrypt them for ransom, for example — the HeartSuite kernel can prevent this access because it doesn't exist in the APO record.

Which means that in order for malware to run and cause damage, a system administrator would need to first add an APO record to allow it to run, and then add file and directory permissions that would allow it to cause damage. The Lockdown feature, discussed below, prevents attackers from adding and modifying APO records.

The bottom line is that each program can be controlled with fine granularity in terms of Permission Orders — in order to allow as much or as little access to files and network locations as desired. And because HeartSuite Permission Orders don't affect physical storage, they can readily overlap — which permits programs to readily share resources, such as shared libraries, without having to make copies and worry about synchronizing updates.

And here's another great benefit: because application of Permission Orders is implemented in the modified kernel, HeartSuite can't be circumvented by any program or user — including the root user!

❤ Core feature #2: Monitor and Denial Modes

Initially, HeartSuite starts out in Monitor mode, which helps you learn how programs behave in terms of access. While it's in Monitor mode, HeartSuite doesn't prevent any programs from running, nor does it prevent access to any files or network connections. Use of the logs during Monitor mode expedites the building of the APO database. Once you've completed the database for the programs that your server runs, you'll switch HeartSuite to DENIAL mode, which prevents a program from running if no APO record exists — and if one does exist, from exceeding its APO record configuration.

You can also use these modes to test untrusted programs. By loading such programs onto a HeartSuite test server and adding them to an APO record without any permission orders, you can review the HeartSuite log to determine whether the program performs only legitimate functionality or contains embedded malware. In fact, this process can even be run in DENIAL mode to protect the test server.

❤ Core feature #3: Lockdown

Lockdown protects the integrity of the APO records by preventing any changes to them. Lockdown is initiated with a simple command and can't be turned off while the server is running, thereby preventing all attacks against HeartSuite itself. If you wish to make any changes, you can simply reboot the server, which always turns Lockdown off.

After you're done, you can turn Lockdown on again with the same simple command. However, a best practice is to enable the HeartSuite startup script to initiate Lockdown automatically, immediately after the server is booted. In that situation, you only need to boot to the original Linux kernel to turn Lockdown off. The HeartSuite installation routine leaves prior Linux kernels in place, recognizing that there may be times when you wish to boot to them.

Once you've made your changes, simply reboot your system and HeartSuite will be restored to Lockdown status, as the default kernel in the GRUB menu. Notably, you must have either physical or serial port access to your server in order to reboot to the original Linux kernel — which means that attackers can't remotely reboot to the original kernel, thus providing another layer of defense.

❤ Core feature #4: File Backup and Versioning

The first files that attackers routinely attack today are your backups. So HeartSuite makes sure to wall them off with extra protection. For starters, HeartSuite automatically backs up files in designated directories, including their subdirectories. But more importantly, HeartSuite prevents access to the backups by *all* programs — only HeartSuite can access the backups. HeartSuite supplies a backup configuration manager tool that admins can use to add and remove directories from the list of directories that will be backed up automatically.

By means of this simple but powerful approach, HeartSuite prevents all programs from destroying or modifying backup files. The HeartSuite version manager program can then be used to retrieve any version of the file at will.

All of that means that if your staff downloads malware masquerading as legitimate software by mistake, the layers of Application Permission Orders, Lockdown, and versioning will work together to minimize damage and easily let you restore your files.

❤ Core feature #5: Ability to Protect Interpreted Programs

APO records can also be created for interpreted code, such as Python, PHP, and Java. In particular, the code containing the main procedure is treated the same way as any ordinary executable program by HeartSuite. HeartSuite relies on programs supplied with it, known as shim programs, to identify the code file containing the main procedure when the corresponding interpreter program (such as Python, PHP, Java, etc.) is launched.

No other technology provides the granularity of control with respect to file and network access that HeartSuite does. No other product provides such an all-encompassing security architecture that combines elements of administrative control with kernel-level implementation.

HeartSuite: Always On the Beat

These five core features are the essence of HeartSuite's groundbreaking approach to security. Of course, there's plenty more going on underneath the hood of HeartSuite — and we'd be glad to answer any technical questions you might have, or discuss HeartSuite in greater detail. Feel free to get in touch with us to ask questions or set up a further conversation.