Ceedo

High-level
Technical
Whitepaper

Ceedo Secure Isolation Solutions

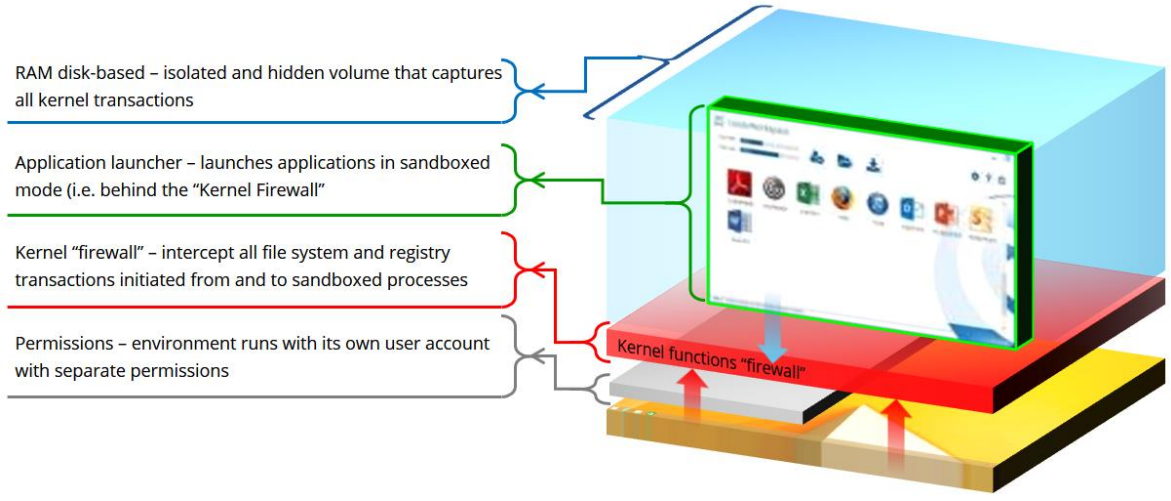
Deep-level multi-tiered application isolation and data privacy solution, allowing secure computing and safe browsing **to and from** non-secure locations with a natural user experience, without relying on separate bootable OSs, Virtual Machines, remote execution servers or signature databases and threat detection systems.

CEEDO'S ISOLATION BASIC CONCEPT

Ceedo's isolation technology provides a desktop security solution that uses a composite of isolation, enforcement and access control mechanisms aimed at secure browsing, safe computing, and privacy.

Ceedo creates a kernel-mode "firewall," providing a deep-level sandbox for applications launched by the user, making sure that anything these applications try to do stays completely isolated from the PC with no possibility of affecting it in any way, while also hiding the data from potential malware that may be active outside sandbox on the host machine in case it has been compromised, protecting both user privacy and underlying security at the same time.

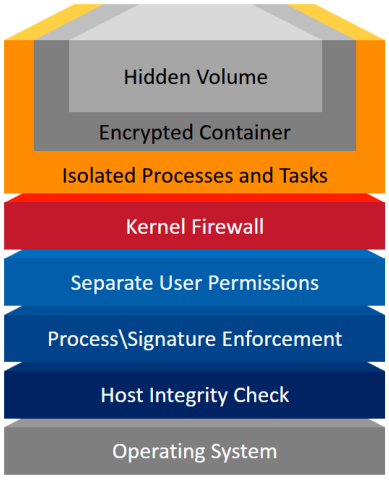
Ceedo's isolation technology provides a powerful firewall-like solution that instead of creating a barrier between a computer and the network, creates a barrier between application operations and the operating system, keeping the operating system, data, and applications separated at a logical level without affecting user experience, bridging the security VS productivity gap.



When an isolated application is launched, all of its read, write, and executed transactions are diverted into an isolated volume, completely hidden from the system, thus providing at the same time, both a "bulletproof vest" for the operating system to deflect attacks, and an "invisible cloak" for session-generated sensitive data, protecting it from prying eyes, and even allowing entire computing sessions to be wiped out at the end of the session, leaving no trace behind.

CEEDO'S SECURITY SOLUTION COMPONENTS

APPLICATION LAUNCHER



The Application Launcher is the main interface between the user and the Kernel Firewall system components. It relays user actions and configurations to the various components (services, drivers and executables), and the most apparent of all – launching applications in isolated mode.

The Application Launcher, uses shortcuts embedded in a launch window, or integrated into the host's desktop and start menu, that launch applications that are natively installed into the host machines (e.g. C:\Program Files\...App.exe), as well as applications that were captured inside Ceedo's own isolated "Layer" (e.g. Layer\Program Files\...). These applications can also be deployed by an administrator, allowing Ceedo to deliver complete isolated environments, including the applications it's supposed to run.

KERNEL FIREWALL

At the core of Ceedo's isolation technology is a component called the "Kernel Firewall." The Kernel Firewall is made up of a set of specialized drivers that are sandwiched at different locations between the user request for an object and the Kernel component that executes said request. For instance, when a user application sends a 'write file' request to the Kernel, before reaching the Kernel itself, the request first passes through the drivers which constitute the firewall.

When a shortcut in the Application Launcher is clicked, the Application Launcher initiates the process belonging to an application and embeds a unique identifying flag that is then inherited by the requests the application sends to the Kernel, allowing the Kernel Firewall to recognize a request as belonging to an isolated application by searching for the embedded flag inside the request itself.

Once a request has been intercepted and positively identified, the Kernel Firewall then determines, based on various rules, if and how it should manipulate the request. Following through on the example of a write-file request, if an application, for instance a browser, tries to write a file to path 'C:\...\Downloads\File.PDF', the Kernel Firewall intercepts the call and can replace "C:\" with some other drive letter, for example, "R:\," resulting in the Kernel executing the write request to "R:\...\File.PDF." After the request has been executed, the Kernel Firewall then takes the altered request and reverts it back to its original state, making the entire system oblivious to any change that took place in between.

While the example above is specific to File System objects, a similar procedure takes place when an application tries to access the Windows Registry, and even when a process tries to launch another child-process, for instance, a browser downloading a PDF document, and then trying to launch Adobe Reader in order to open it. In this case, the Kernel Firewall will wrap the new process (i.e. Adobe Reader) inside the task-isolation sandbox, together with the process that launched it (i.e. the browser), making sure that Windows interoperability is retained for the best user experience, but still keeping everything running behind the Kernel Firewall, including processes, child processes, shared objects, data, etc.

In terms of security, what the example above means is that if the PDF file the user has downloaded had been infected with some form of malware aimed at corrupting the underlying system, as in the case of ransomware, not only will the PDF file itself be segregated from the system, but so would the process that was launched to open it, as well as any action it may have tried to perform, down to the very last kernel transaction.

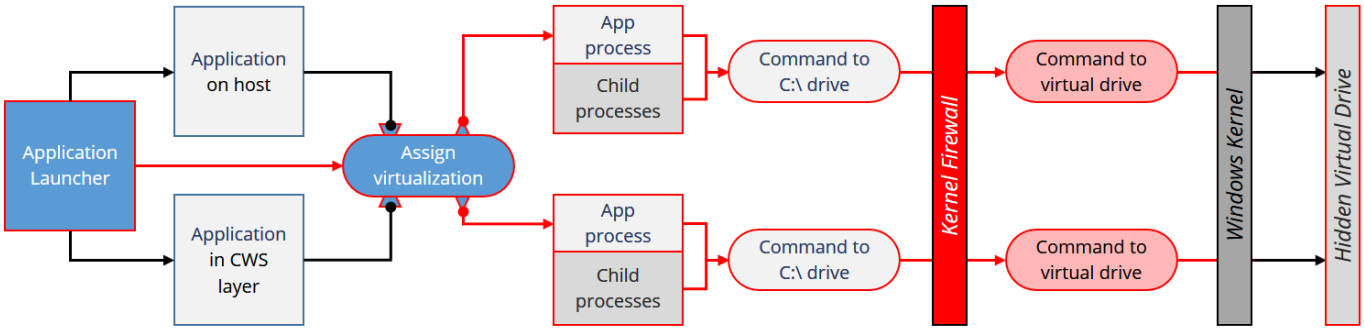
VIRTUAL DRIVES

Once a call to a process or task has been intercepted by the Kernel Firewall, it can decide to redirect that request to a different volume (or hive), as explained above, or to allow it to fall through to the host (usually in the case of read and execute requests). In most cases, if a write request originated from an isolated process, it will be diverted to the virtual hidden drive, unless it was specifically excluded or blocked entirely by the permissions engine (see below).

The Virtual Drive, also referred to as *Layer*, is a Virtual Hard Disk-based volume (VHD) that is mounted into a device – but not assigned a drive letter – that uses a proprietary interface not recognized by Windows as a hard disk, or accessible via any traditional means, so processes running outside the sandbox are completely oblivious to the drive's presence, and cannot reach the data stored in it. This means that any sensitive data that may be stored on that drive is completely hidden from anything running outside the sandbox. Moreover, in most cases, the hidden drive is temporary, or even resides solely in RAM as a RAM-disk, meaning that once it is unmounted, no information is left behind on the machine.

The VHD files from which Ceedo's Layers are constructed, can be stored on the web and downloaded by our Application Launcher system to the local machine, or they can reside on a network drive with a local child disk, or even on a removable drive.

Additionally, Ceedo supports having more than one virtual drive attached to the system. This allows for an extra drive for user generated data if one is required to persist across sessions. In some cases, a drive might be dedicated to applications that were virtualized via Ceedo's isolation, or with 3rd party tools.



PERMISSIONS AND ENFORCEMENT ENGINES

CeedoWorkspace comes with a few additional protection mechanisms to ensure that the computing environment meets all the minimum requirements that the user or administrator have set.

The first mechanism is based on Windows' own built-in NTFS and user access control permissions. For this, Ceedo creates a temporary virtual user account – CWSuser – and runs designated applications under this account, ensuring complete separation between the isolated applications and the rest of the desktop environment at the user session and NTFS permissions level. This allows applications to have their own read and write permissions, network and resource access, etc., which differ from that of the logged on user. For instance, banning the sandboxed applications from reading from or writing to network drives, printing documents, and performing any type of unauthorized changes, all of which achieved by simply running the CWSuser as a restricted and limited user account, even if the logged on user is an administrator.

The second mechanism is a process and signature enforcement engine that, based on various rules, can prevent any unsigned process from running, stop processes with specific certificates from running, stop specific process names from running (with and without MD5), and more.

The third and last mechanism is an integrity check for both the host and Ceedo's internal files. This includes running over all of Ceedo's own components, making sure that all of its DLLs and executables are signed by Ceedo, and additionally, an MD5 check runs on all the Ceedo Layers. This makes sure that if a VHD has been tampered with, it will not be used, and a fresh copy of the VHD will be downloaded from the server. Lastly, Ceedo's isolation solution also comes with a host health-checker, ensuring that the host itself meets minimum requirements such as OS version and patch level, antivirus and firewall status, specific VPN status, and more.

ENCRYPTED VOLUMES INTEGRATION

On top of its internal Virtual Drive system, Ceedo has a deep coupling with VeraCrypt (VC) encrypted volume software, which allows the VHD files to be stored inside a VC-encrypted container. The VC container itself is created on-the-fly on the client machine, and when applicable, if an administrator pre-assigned specific Layers to the user, they are downloaded directly into the encrypted volume. Furthermore, to make sure that the VC encrypted container cannot be opened by a different user and/or on a different machine, or opened directly through VeraCrypt itself, the passwords users provide for their own container during creation, are salted with various machine and user-specific parameters, so even with the correct password, the VC container cannot be opened if it is not accessed directly from Ceedo's interface on the specific machine it was installed, and by the specific user that installed it.

COMMON SCENARIOS

SAFE BROWSING AND SECURE COMPUTING

Ceedo provides the ultimate solution for safe browsing, protecting the PC from harmful files that might be intentionally or inadvertently downloaded and executed by the user, especially if said files somehow manage to bypass the detection of a local antivirus or similar program. For instance, a click-fraud introducing some ransomware into the system, a botnet hijacker infiltrating after a drive-by-download, malicious code hidden in documents, and more, all stay confined within the task isolation sandbox, and wiped out at the end of each session when the environment is shut down.

In addition to protecting PC, by launching browsers and any of their child processes inside a task and data isolated runtime, Ceedo can be configured to always wrap an insulation layer around any browser the user launches, even if the Application Launcher was not used, or initiate an insulation layer whenever the user tries to save or launch a file from an email client.

SECURITY AND PRIVACY ON COMPRMISED MACHINES

In many cases, users and companies wish to keep certain computing sessions private, either when a user is using a website hosting potential threats, or when contractors and employees need to access organizational resources from unmanaged machines that might be compromised. In these scenarios, users and administrators can decide to not only secure data behind the Kernel Firewall and keep it concealed within an inaccessible hidden drive, but also to use the Ceedo's enforcement, access control and host checking mechanism. For instance, not allowing the system to run on unpatched Windows machines, denying unsigned processes from running during the use of Ceedo, and of course, running the isolated applications not only behind the Kernel Firewall, but also, under a different user account with strict permissions that deny information exchange.

USER-INSTALLED APPLICATIONS

Maybe one of Ceedo's most unique abilities is that fact that, if configured to do so, it allows users to install their own applications within safe and isolated environments, even if they are limited users running on highly restrictive desktops.

Using Ceedo, users can install applications under the CWSuser account, completely isolated from the system without any option of making actual changes to the underlying image, but still allowing them the computing freedom they require in order to stay productive in today's modern workplace.

The task isolation wrapper provided by Ceedo, can capture applications and make them accessible to the user without exposing the computer to outside threats, without making any changes to the desktop's actual composition, and allow administrators to completely lock down their systems, all without hindering user productivity.

CONCLUSION

As isolation emerges as the most important technology to be introduced into the realm of cyber security, due to its abilities to protect users and organizations from zero-day attacks, unknown exploits and privacy issues, Ceedo's groundbreaking isolation technology finally bridges the gap between security and user experience with little to no investment in infrastructure, without adding management overhead or requiring users to change the way they normally work, while providing an unprecedented level of isolation at the deepest possible level of the OS, spanning from data to applications, and complemented by sophisticated tools that are designed to provide complete secure and isolated computing workspaces for users.

Using CeedoWorkspace, companies and users alike can ensure that their PCs are kept safe and secure from unknown outside threats originating from compromised online resources, while allowing sensitive computing sessions and data to be kept private and secure on potentially compromised machines, and do so without introducing new hardware into the datacenter, and without relying on cloud services or Virtual Machines, without affecting Windows' interoperability and workflow, which is crucial for productivity.