CLOUDPAGING SECURITY

October 2017

CLOUDPAGING SECURITY BACKGROUND

The security technology behind Cloudpaging has been battle tested for more than a decade on premises and over the Internet and remains unbroken. It has delivered and continues to deliver some of the most commercially expensive, complex, and desirable software applications. Cloudpaging provides a digital rights management ("DRM") security model to protect applications from anti-piracy with or without existing software security, and is fully integrated with our delivery and virtualization technologies to synergistically provide stronger and more sophisticated application security than traditional DRM wrapper mechanisms or ones requiring re-compiling of target applications. Its implementation includes data compression and encryption, protocol and caching encryption, file and resource level access control protection, and DRM. They combine to provide the best protection for online software applications. In addition to its inherently secure foundation, Cloudpaging's DRM is quite feature rich. It includes monitoring, metering, auditing, enforcement, and integration capabilities with other authentication and commerce systems to enable commercial and enterprise Appstore experiences for a target application's audience. It also automates the push delivery of applications through similar mechanisms for remote desktop management in enterprises. Cloudpaging provides integration with other security components by use of REST APIs, SOAP, and SSL/TLS. Every attempt has been made to make the security system transparent to a privileged end user through the use of integration and advanced capabilities. For example, integration with single sign-on, Active Directory or existing authentication mechanisms reduces the learning curve of using Cloudpaging, and usage and activation based licenses allow for flexible options for the end user of online and offline time limited models, respectively. Monitoring, metering, and auditing capabilities are accessed through web-based user interface

and are also logged into a standard SQL database where custom queries, audits, and reports are generated with existing and standard tools. Administrators and ISVs have access to concurrent (pooled) or per seat license controls, and can count seats depending on process usage (actual use of the application) or when the end user is ready to run the target application on his system. Enforcement is further applied with platform and version constraints which can be placed on end users for target software applications, including which target operating end users are allowed to execute on and the ability to force end users onto a specific version of the target application. Below are key features of our flexible, security architecture:

- Virtualization is fully integrated and leveraged to provide security beyond traditional mechanisms (via architecture) and application security;
- Management of applications and license policies is provided for applications with a web-based user interface;
- Robust encryption is used, including AES 256 on containers containing Cloudpaging applications and in any local end user device cache
- Multiple license policies can be used to control application seats and usage metering, or controlled roll outs to groups;
- Administrators know exactly how many application seats are in use and historically how long each user requires a seat;
- License policies are always enforced, regardless of the client machine being online or offline;
- The same trusted web security mechanisms that financial industries rely on for both authentication and encryption of software content delivery over the Internet is implemented;
- The patented DRM is novel, as is the delivery and virtualization mechanisms;
- Works with most existing ISV's license mechanisms, including dongles and license keys;
- Allows for and manages on-line and off-lining of target applications;
- Allows control of the platforms that the application can run on e.g. Windows OS's
- Active Directory support: Any licenses can be tied to an Active Directory object, which allows for named and group associations;

- Can layer over OS components, which can be configured to prevent use of unauthorized software (even installed ones);
- Token based security easily integrates into third party mechanisms to prevent unauthorized users access to the system;
- Provides an extremely scalable security model since up to 40,00 users can use a single DRM server; and
- Cloudpaging delivers patches to applications faster, thereby reducing delivery time for critical security updates.

SECURITY ARCHITECTURE

Distributing digital content can be an inherently insecure exercise. However, Cloudpaging provides a sound and secure foundation for deploying valuable and private software assets onto remote machines. Cloudpaging employs several security principles in its architecture: strong communication, strong data encryption for all persistent storage in the client and server, and strong (and periodic) coupling between the client and server to reassert continuous control with the client. It allows users to access the application in its entirety without ever installing the full application and this creates a strong coupling between the client and server. This combined with patented DRM ensures a risk-free method of delivering software.

From a technical requirements perspective, Cloudpaging prevents code from being captured or corrupted in transit while ensuring downloaded portions of application run only with valid licenses issued. It also allows management capabilities for the administrator to control scaling and various license configurations at runtime. In other words, Cloudpaging provides copy protection, license enforcement, and management abilities:

- **Copy Protection**. With Cloudpaging, it is impossible for anyone to obtain a clear-text copy of an entire protected application. This includes both illegitimate access by hostile attackers and preventing legitimate users from making an unencumbered copy.
- **License Enforcement**. Cloudpaging quantifies time, usage limits, operating system, version, online and offline limits and policies.

- **Central management of application deployment and licensing**. Cloudpaging dynamically controls and tracks application usage and user behavior while preventing licensing infringements and complying with licensing agreements.

The remainder of this paper covers the technical solution associated with Cloudification and Cloudpaging container security which define the security parameters of the assets, and then explains how the system enforces security through its services and execution protection.

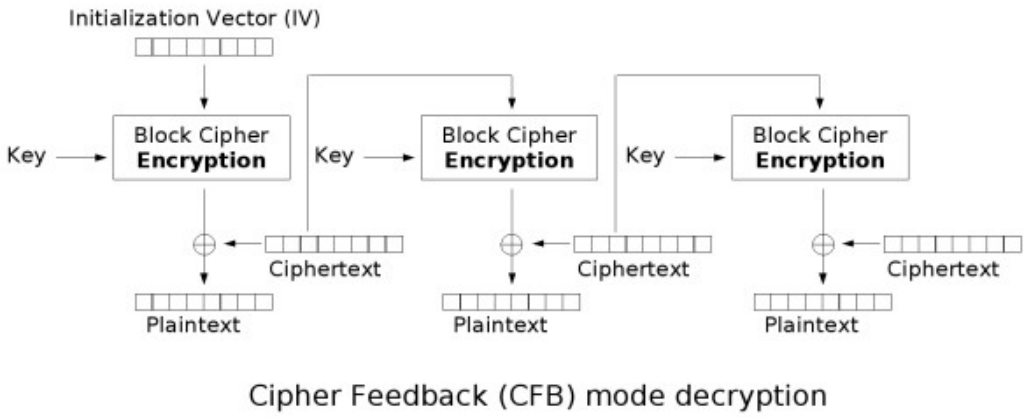
CLOUDIFICATION – CREATING THE APPLICATION CONTAINER

The first step in creating an Cloudpaging Application is called Cloudification, the process of creating of an appset container. The Cloudification process, or Cloudifying, records all of the changes made to the system by installing the target application, configures the recorded application, and then enhances the application set through the Cloudpaging virtualization and delivery parameters. That process generates the application set which is embodied in a single container file. This container can then be loaded onto the server and Cloudpaged to the Cloudpaging Player with the required token to execute the application, or opened directly by unpacking the Appset file into individual parts which includes the token and other content. The end user can then launch the target.

APPSET SECURITY

Appset security refers to the content protection of an application package being stored or transmitted. Cloudpaging provides this protection through encryption of the application's data blocks and configuration data which consists of mainly static data. The encryption method used in Cloudpaging is Advanced Encryption Standard ("AES"). This encryption method has been adopted by the US government and approved by the NSA for encryption of classified and top secret information. For this reason, Cloudpaging has been registered with the US Bureau of Industry and Security so that the export of Cloudpaging to other countries is permitted (with exceptions).

AES encryption in Cloudpaging with 256-bit keys by default, which provides the highest level of protection. Cloudpaging utilizes CFB (Cipher Feedback Block) by default (see **Figure 1**). Using CFB provides the benefit of low latency between the arrival of plaintext and the output of the corresponding ciphertext, which can be important for Cloudified applications if encryption is to be done during transmission. In addition with CFB, decryption can execute in parallel and 1-bit corruption in the input ciphertext affects only two adjacent blocks in the output plaintext. Cloudpaging also leverages hardware AES to further accelerate performance.



Cipher Feedback (CFB) mode decryption

Figure 1

VIRTUAL SECURITY

Virtual Security refers to the protection of virtualized resources within the Appset or target application. A protection setting is specified individually for each resource though not all resources can be controlled with the same level of granularity. For example, files have security settings and can be assigned to one of four layers (two are protected), whereas fonts and environment variables are either protected or accessible only to the privileged application, or unprotected.

The layers mechanism provides virtualization of key resources which enables special access and copy protection. Process, layer, and additional rules put in place for virtualized resources determine if the resources are accessible and how those resources will behave when an attempt

is made to access that resource. For example, when a file is set as isolated, it is accessible only to the application process for which the file was packaged.

Layer	Protected?	Description
Installed-permanent (Layer 1)	No	Copies assets (files, folders, registry keys, and registry values). Installed permanently onto the local system, and can be seen by the entire local system
Installed-temporary (Layer 2)	No	Installs assets during the activation process, and uninstalls assets during the deactivation process. The original asset is backed up before the new asset is installed, and when the new asset is uninstalled the original asset is restored
Virtual-integrated (Layer 3)	Yes	Assets that can be seen both by the paged application and the local system, but are not physically installed on the local system. This feature helps copy protection aspects.
Virtual-isolated (Layer 4)	Yes	Assets that can only be seen by the paged application, but are not physically installed on the local system. This feature helps copy protection aspects, more so than Layer 3 by further restricting access to the resource.

Table 1

Four layers are currently provided as listed in **Table 1**. In the context of anti-piracy, the virtual layers provide copy protection.

For file resources, Cloudpaging provides further controls in addition to the layers mechanism to aid copy protection. Files can be set with four additional security settings as listed in **Table 2**.

Name	Cloudpaging Studio Name	Resource Type
Deny-write	Deny Writing and Modifying	File
Deny-search	Hide from folder listings	File and Folder
Deny-cache	Do not cache on local machine	File
Deny-copy	Deny Reading and Copying	File and Folder

Table 2 – Securities Vs. Resource Types

Details of those security settings are as follows:

- **Deny write**. The file is read-only and its content cannot be overwritten. The restriction applies to Cloudpaged applications and all other processes in the local machine. Unless overridden, this security attribute is enabled for all .exe files.
- **Deny search.** The file is hidden from the file listing of its parent folder. This restriction applies to the Cloudified application and all other processes in the local machine. Unless overridden during the Cloudification process, this security attribute is enabled for .exe files. Note that this security only protects against directory listing and it does not prevent access to the file. If a process (Cloudpaged or not) knows the absolute path to the file and if the process is allowed to access the file, it can get to the file directly without first finding the file in the directory listing.
- **Deny cache.** The file content is not to be cached in the local machine. Any page request of the file must go to the Server. This restriction applies to the Cloudpaged application and all other processes in the local machine. This security attribute is not enabled by default but it is useful in the case where the Appset is physically distributed to user machines but the administrator wants certain blocks from the Appset stripped off and made available on the Cloudpaged service only.
- **Deny copy.** The content of the file cannot be copied. Additionally, the creation of any empty file with the same name as the file or with a name in the form "Copy of <thefile-name>" or "<the-file-name> Copy" is prohibited. This rule applies to both the Cloudified application and all other processes in the local machine. Unless overridden during Cloudification, this security attribute is enabled for .exe, .dll and .ocx file.

Hypothetically, nearly all virtualized resources could be set with the maximum security setting including setting all resources to isolated mode, Layer 4, but doing so may limit application functionality. Help systems or resources which require third party applications or processes, including the operating system itself, will need access to the application assets for the application to behave properly, or to run at all. However, this too could be handled through process override method which allows certain processes, identified by name, path, or hash to access restricted resources within the Appset. On the other hand, in some circumstances it is perfectly acceptable, if not desirable (e.g. for the purpose of running multiple instances of the application on the same operating system), to restrict access to all virtualized resources. For that purpose, layer 4 (isolation mode) provides that functionality with maximum security.

SERVICES

The Cloudpaging server delivers the target application adhering to the package security and the preconfigured content definition of the Cloudified Application covered in the two prior sections. There are two main services of the server: The Cloudpaging streaming service and the license service.

The license service is responsible for granting access to applications by generating tokens. Tokens, which are passed between the server and client are generally a few kilobytes in size, and contain all rights required, including location of Cloudpaging services, to execute an application on the Cloudpaging Player. The token itself is also encrypted and signed with an asymmetrical key to protect data and to securely authenticate. Tokens are delivered over SSL/TLS to further encrypt the transmission between the client and server. The sessions are very small, independent, unique and relatively infrequent which allows this service to be very scalable.

The Cloudpaging streaming service is responsible for Appset delivery. It stores a copy of the Appset and delivers fragments to its destinations pre-encrypted. The Player ultimately depends on this Cloudpaging streaming service to deliver Appset resources that are not available in its cache. To support high-volume scenarios, separating the Cloudpaging streaming services from the license service and distributing them is ideal. Optionally, the Cloudpaging streaming service

can be delivered over SSL/TLS to further encrypt the transmission between the client and server though this impairs scaling properties of using proxies and Internet CDNs.
The Cloudpaging streaming service also provides a sub-service: the heartbeat. The heartbeat forces the Cloudpaging Player to continuously check in with the Cloudpaging streaming service at the frequency programmed by the administrator. If the heartbeat is unable to contact the Cloudpaging streaming service, Cloudpaging has the ability to dynamically restrict access of an end user to an application independent of the license service within a specified time (See **Figure 2**).

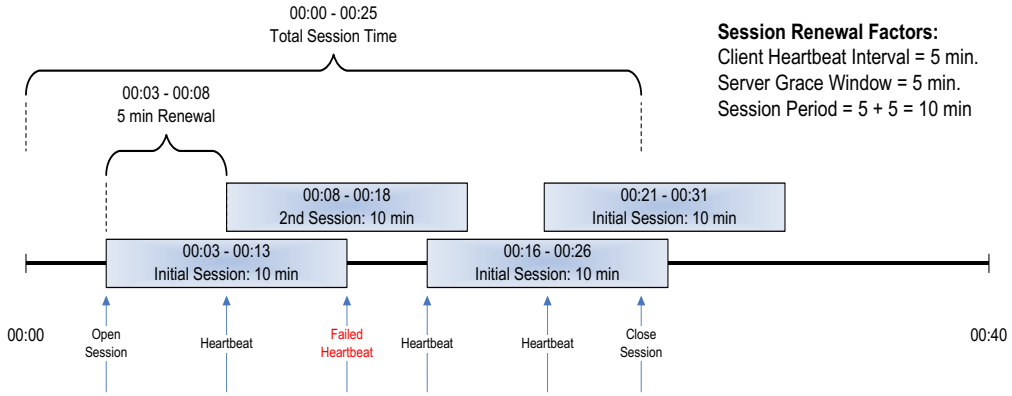


Figure 2 – Heartbeat Sequence

EXECUTION PROTECTION

By combining the processes and techniques above, the Cloudification process can configure key parts or all of the application to be protected and invisible to all (including the operating system) but the privileged executing processes. The Cloudpaging Player, a privileged kernel level application that intertwines closely with the operating system, ensures that all of these policies are enforced during execution of the application by utilizing its key fundamental technologies, virtualization and Cloudpaging to enforce and provide most aspects of its security. Therefore, secure execution relies on both Cloudpaging Server and Player.

During Cloudpaging Player execution, virtualization plays as the key mechanism for Cloudpaging security by making Appset resources only available to privileged resources and processes,

hiding resources to all but itself, or making a resource appear as if it were there but in fact it is not. By working with the underlying operating system's persistent mechanisms and environment, Cloudpaging intercepts each and every request in the virtualized space to ensure that the user's current license is still valid and that the resource is being accessed by privileged processes and not by any other program. Until that point, Cloudpaging keeps any portions of the application that have already been Cloudpaged in the encrypted cache. The Cloudpaging Player is the only mechanism that contains the secret key necessary to decrypt the cached application data. The symmetric key is stored in the Cloudpaging Player's proprietary keystore, and every key and thus cache is unique for every Cloudpaging Player instance.

This approach is part of a general family of split-horizon security solutions: Effective operations of the protected code on a computer that is not trusted must be made to depend on some (unpredictable) output from another, trusted computer. The security of any split-horizon approach depends on the unpredictability of the interaction that crosses between the not trusted and trusted portions of a system. In the case of Cloudpaging, the unpredictable interaction is between the Cloudpaging Player and the Cloudpaging Server and its services. Cloudpaging streaming is inherently an unpredictable element of interaction, primarily driven by the end user who randomizes the use of the application. Though the heartbeat is predictable, it is protected and serves as another dependency to strengthen the split-horizon approach. Other random infrequent functions such as upgrades serve as further measures that add to security.

Though there are many steps to the security process, the end user experience is transparent. From their perspective, installing a Cloudpaging Player-protected application is seamless since the initial download is replaced by provisioning the relatively small Player and a few desktop shortcuts for the new application.

After this point, each time a new Appset is authorized for a user, Cloudpaging Player virtualizes a new virtual space with all of assets as specified and virtual resources to make it appear that the application has been installed locally. The only difference is that on any attempt by any Windows process to access a resource from the virtualized space will require a renewed license token

from the license service to be generated and then decrypted by the client periodically. If not, the Cloudpaging Player will attempt to terminate the application sessions, after a warning and a grace period for the user to save any open files, etc.

CONCLUSION

Cloudpaging combines many forms of proven security techniques, mature mechanisms, and innovative proprietary methods and mechanisms to ensure that applications are distributed as securely as possible. Please contact a Numecent representative to find out more about distributing your applications on-demand with full confidence.