



## Vuuch Security white paper

### Overview

Today, most modern applications are delivered via what has come to be called “the cloud.” Cloud computing means many things and is used loosely to define everything from Gmail or Microsoft Exchange email stored on a rented server to very technical application programming interfaces designed for developers only. Because there are so many different definitions of what cloud computing is, it is important to know how an application is designed and implemented so its security measures can be assessed.

This document will describe at an architectural level how Vuuch is designed and implemented in the cloud. The document also discusses specific security, availability and reliability features that customers can rely on when considering how to deploy Vuuch while complying with their internal security requirements. Vuuch has been designed and implemented to be highly secure, scalable and reliable while providing all the cost-savings and reliability of a cloud-based implementation.

### How Vuuch’s Architecture Enhances Corporate Security

Because of the popularity of the term, many technology companies want their applications to be perceived to “run in the cloud.” And because there is no standard definition of how a cloud computing app should be designed and architected, most companies can legitimately claim to be “in the cloud” if they provide any services whatsoever using the public Internet. Even the traditional IT data center is recasting itself as a provider of “private clouds.”

Vuuch takes a much narrower view of what a cloud application is. To us, a cloud application has to take architectural advantage of cloud infrastructures, such as multi-tenancy and encryption. And it must feature an internal design that focuses on security. When evaluated against these more rigorous definitions of a cloud application – as opposed to a purely marketing definition of cloud apps – we believe users will conclude that Vuuch leverages cloud computing to actually make Vuuch *more* secure than typical web applications. Here are some of the architectural and implementation details that enable Vuuch to be more secure.

***Vuuch is designed from the ground up for multi-tenancy.*** Multi-tenancy is a complicated topic, but it fundamentally describes an application that is able to securely separate multiple users of the application from each other while they are using the same system resources. Gartner, the IT analysis firm, has written extensively on the topic of multi-tenancy, but you must be a client to access their reports. However summaries of their recommendations are available in blog posts. Yefim Natis, an analyst covering cloud computing, accurately summarizes the core attributes of multi-tenancy in this blog post: [http://blogs.gartner.com/yefim\\_natis/2010/03/24/on-multi-tenant-elasticity/](http://blogs.gartner.com/yefim_natis/2010/03/24/on-multi-tenant-elasticity/). Compared to this set of specifications, Vuuch delivers multi-tenancy and with it, architectural separation of client data.



Vuuch runs on Amazon's EC2 (Elastic Computing Cloud) infrastructure. EC2 provides security, scalability and elasticity for Vuuch. You can learn more about EC2 here:

<http://aws.amazon.com/ec2/#functionality>.

With EC2, Vuuch takes advantage of an infrastructure of essentially limitless resource. EC2 data centers are more secure than many corporate data centers, from both a physical and network intrusion perspective. Vuuch implements EC2's network firewalls to prevent distributed denial-of-service (DDOS) and cross-site scripting (XSS) attacks, among common attack vectors. Using EC2 availability zones and other EC2 features, the attack surface Vuuch presents is dramatically reduced. The many implementation, failover and backup options EC2 offers actually *enhance* Vuuch security because they contribute to the resiliency of Vuuch. It is easier to stay ahead of the bad guys when your system is less prone to attack from the network and is running securely inside a world-class data center.

***Vuuch is architected to provide true user isolation.*** As another Gartner blog post points out ([http://blogs.gartner.com/neil\\_macdonald/2011/01/14/multi-tenancy-doesnt-have-to-be-bad-for-security/](http://blogs.gartner.com/neil_macdonald/2011/01/14/multi-tenancy-doesnt-have-to-be-bad-for-security/)), "As long as the multi-tenancy mechanism of the application does what it says it does in terms of effecting separation of the different tenants, it really isn't all that different from the virtual machine scenario." Virtualization is generally recognized as a secure method of sharing system resources with complete isolation of one virtual machine from another. Vuuch has implemented multi-tenancy within the system to achieve the level of equivalence Gartner describes.

Internally, Vuuch implements a domain concept. Domains prevent one company's information from being accessed by any other, unless they are given permission to do so. In addition, Vuuch allows a user in a domain to be specified as a domain administrator, giving a company complete control over its users and their access.

***Vuuch resists social-engineering attacks and does not require access to sensitive files to permit effective team interaction.*** Most legacy collaboration systems provide access control lists (ACLs) and/or file protection schemes in the form of data access controls in an attempt secure user access. Both are surprisingly subject to attack. ACLs are complex and administratively costly. Worse, the loss of a single master password can result in the loss of *all* security. File-based permission schemes require significant end user education and are very difficult to implement as universal end user compliance is required for successful implementation. Both ACLs (via their password schemes) and file-based permissions are subject to social engineering attacks ("phishing") in which users can be persuaded to share information which criminals can use to attack a legacy collaboration system.

Vuuch is neither ACL-based nor file-based. Users, by default, have access only to their content. Users grant and revoke access to information using the social concept of "inviting" and "disinviting" users to view content. In this way, users naturally secure information because they are indirectly classifying information when creating or modifying that data. Internally, Vuuch rigorously enforces users' invitations to information and removals of access from that information, ensuring that the original

security intent – who is (or was) allowed to see this? – is preserved. In Vuuch, users comply without having to think about compliance because Vuuch’s architecture automatically enforces information security. This makes following corporate information security policies natural for non-technical end users and therefore enhances compliance. Every change to a Vuuch page or activity is logged and available as in an event stream for audit or reconstruction of the sequence of events.

By design, Vuuch does not impose, alter or restrict an enterprise’s current file management and/or product data management tools and techniques. Once again, for emphasis, *Vuuch does not require access to a file to allow team members to interact on the contents and knowledge in that file.* This concept may take some getting used to because many users and security administrators assume that content must be stored “in a file” and that moving the file around a computing infrastructure is the key to promote collaboration. In fact, files are really just “cold storage” for designs, decisions, tasks and judgments that the team has made. These files only document the final design or decision – not how that decision was reached, by whom, over what time period and at what cost.

Worse, emailing sensitive files (even though they are incomplete) creates an email storm that can lead to a security mistake on the part of an end user, for example sending it to the wrong user. And email is remarkably insecure. By default (and that is how most organizations use email), it is neither encrypted nor authenticated. Companies often spend enormous effort to secure their data and systems, only to lose much of their IP through targeted attacks (“spear phishing” or “weaponized email”) conducted through email. In addition to not changing the way files are processed, Vuuch makes it possible to completely stop using email for interaction with the product development team, customers and vendors. This can significantly enhance overall security.

Since Vuuch does not require files to allow people to work on their deliverables as a team, how, exactly, does work with files? Vuuch “fingerprints” a file – be it a Microsoft Office document or a CAD file – using that file’s published interfaces to store the fingerprint. For example, in Microsoft Word, it is possible to store custom properties in the document. This is what Vuuch uses in Word. In other programs, there are similar capabilities.

A unique “serial number” (and nothing more) identifies the file to Vuuch no matter where or how it is managed. In this way, when a file is Vuuched (that is, when its serial number is assigned) and whenever a Vuuch user legitimately accesses this file via normal security procedures, Vuuch can present to the user the entire set of meta data Vuuch users created and collected about this file. Vuuch only knows a file by its fingerprint; there is absolutely no content from the file stored in Vuuch. Moreover, the fingerprint is useless as a key to accessing any of the file’s contents. For example, if the document is encrypted, the Vuuch fingerprint is useless to decrypt the document because the Vuuch fingerprint is not a cryptographic key. In this way, separation of the most crucial data – the record of the business process used to create the file – is separated from the file itself. That highly sensitive information is secured in Vuuch where it can be managed far more easily than, for example, trying to reconstruct who emailed which file to whom when.

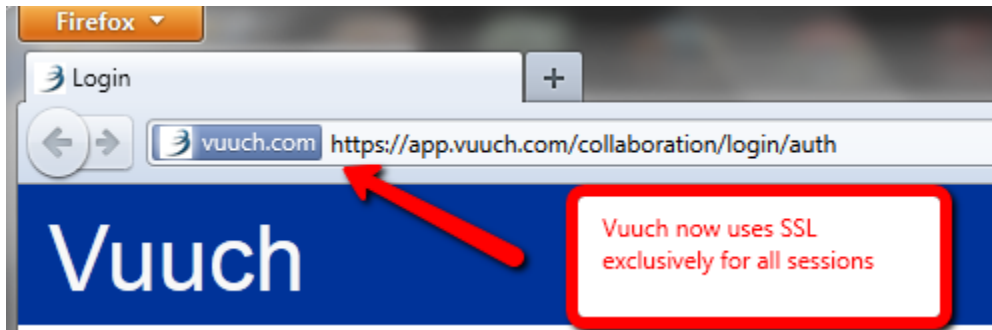
Legacy collaboration systems (like Microsoft SharePoint) actually complicate security because they are fundamentally based on users managing files themselves. Vuuch breaks this link and in doing so, achieves better security along with significantly enhanced user productivity. In other words, users worry less about group editing files and/or emailing them around and instead spend their time doing what they want to do: making decisions, evaluating current and past progress, discussing issues and assigning tasks.

***Vuuch implements best practices to enhance security.*** As with any online system, Vuuch provides a userid and password scheme. Vuuch's implementation of passwords is more secure than many because Vuuch does not store the actual user password. Instead, Vuuch hashes ([http://en.wikipedia.org/wiki/Hash\\_function](http://en.wikipedia.org/wiki/Hash_function)) passwords and stores only the hash. This means that unauthorized access to Vuuch's password store would yield nothing of value to a hacker. In addition, it means that Vuuch itself cannot recover a user's password because hashes are a one-way algorithm. And, because Vuuch hashes whatever password a user supplies, passwords can be up to 128 characters length and contain any combination of ASCII characters. Users may reset passwords only by request to a previously verified email address. With this implementation, enterprises can enforce their existing password schemes in Vuuch.

## Managing Security in Transit

***Vuuch uses SSL for all data transfer to and from clients.*** Good architectural design for the cloud and operational excellence in the data center can still leave information vulnerable to interception during transmission. So-called "man in the middle" attacks can eavesdrop and/or capture data during transfer and assemble a picture of the transmitted data. While this may not be an issue in Vuuch due to the separation of sensitive files from the team interaction on those files (discussed above), Vuuch encrypts all communications between Vuuch and clients using the Secure Socket Layer (SSL). A description of the mechanisms used in SSL is beyond the scope of this paper. A very good introduction to SSL is available from the *Security Now!* podcast. (mp3 audio: <http://media.grc.com/SN/sn-195-lq.mp3> and PDF transcript: <http://www.grc.com/sn/sn-195.pdf>).

Because SSL is such an important security feature, we will describe a few of the Vuuch-specific SSL implementation details to help readers in their evaluation of Vuuch security. First, there is no way to connect to Vuuch without an SSL connection. This includes all common browsers as well as all Vuuch plug-ins, including those for Microsoft Office, SolidWorks, SpaceClaim and other CAD systems. Second, 100% of the transmission is encrypted. Some systems encrypt only the password entry and then return the user to a non-encrypted session. (Facebook and Yahoo! email are examples of this behavior, though Facebook now has a setting to enforce 100% encryption.) One reason systems have avoided 100% encryption has been server overhead. Vuuch has addressed this problem by using a very high-speed web server to encrypt and decrypt all communications before transferring the data to or from the Vuuch processing server.



Third, SSL offers the end user confirmation that he or she really is communicating with Vuuch and that there is no man in the middle – or if there is one, that attacker is seeing encrypted data that might take centuries of brute-force key guessing to decrypt. Browsers vary in the way they present this information, but a typical representation is shown above in Firefox 4. The authentication capability of SSL enhances security by allowing users to make certain they are communicating with a valid Vuuch server at all times.

## Summary

No matter where an application runs – in the cloud, on a local machine or in a corporate data center, good security is a mixture of operations, architecture and knowing how users work. Vuuch has addressed all of these areas in its native cloud implementation and believes that it has used the most modern and sophisticated designs and processes to deliver a unique combination of cloud accessibility *and* excellent security.

We hope you will contact us if you have any additional questions or concerns. You mail email us at [contact@vuuch.com](mailto:contact@vuuch.com) and/or call us on +1 617 500 8100. And thank you for using Vuuch.