BLUE COAT®

DATA PRIVACY AND COMPLIANCE IN THE CLOUD

HOW ENCRYPTION AND TOKENIZATION SATISFY INDUSTRY MANDATES AND LEGAL REQUIREMENTS ASSOCIATED WITH PROTECTING SENSITIVE DATA IN SAAS CLOUD APPLICATIONS.

BLUE COAT®

Security Empowers Business

INTRODUCTION: WHAT'S THE CONCERN ABOUT PLACING SENSITIVE DATA IN THE CLOUD?

Software-as-a-Service (SaaS) delivered via the public cloud is one of the fastest growing IT technologies being tracked by Forrester. They report that worldwide spending on the SaaS market is expected to grow at the rate of 21 percent over 2015, reaching $106 billion by the end of 2016. It's quite likely that SaaS growth projections could be even higher if it weren't for prospective users' lingering concerns about protecting sensitive data. According to Forrester analyst Liz Herbert, "Security is the No. 1 reason preventing firms from moving to SaaS."

More than anything, these security concerns are born from a series of data residency/privacy and industry-specific regulations that describe how data must be treated in the cloud.

The regulations include:

- Data Residency/Privacy Legislation – laws in specific states, countries or governmental associations such as the European Union (EU) that dictate that sensitive or private information may not leave the physical boundaries of the country or region (residency), and that the information should not be exposed to unauthorized parties (privacy).

  Example legislation includes:
  - The United Kingdom Data Protection Law
  - The Swiss Federal Act on Data Protection
  - Russian Data Privacy Law
  - The Canadian Personal Information Protection and
  - Electronic Documents Act (PIPEDA)

  The EU Data Protection Directive is also an important piece of data privacy legislation that regulates how data on EU citizens needs to be secured and protected. With the European Court of Justice's ruling in 2015 that the Safe Harbor framework is inadequate to protect the privacy rights of EU citizens when their data is processed in the United States, data privacy professionals expect to see additional data privacy legislation and restrictions appear across Europe.

- Industry-specific Compliance Requirements – laws or mandates covering a specific industry, type of business or government agency that prescribe the appropriate treatment and security of private or sensitive information. Examples of these types of requirements include:
  - The Health Information Portability and Accountability Act (HIPAA)
  - International Traffic in Arms Regulations (ITAR)
  - The Health Information Technology for Economic and Clinical Health (HITECH) Act
  - The Payment Card Industry Data Security Standards (PCI DSS)
- Third Party Obligations – agreements among business partners that outline how a party such as a contractor or vendor will handle and treat private or sensitive data belonging to another organization. Such agreements often hold the external party accountable for securing the data in the same fashion as the owner of the data, including adherence to all residency, privacy and compliance requirements. For example, a contracted agency performing billing for a hospital in the U.S. must observe all the data protection requirements mandated by HIPAA and HIT.

In view of all the residency and compliance requirements that companies face, it's a challenge to strike a balance between safeguarding sensitive data and attaining maximum benefit from SaaS applications. It's simply not an option to place clear text data in the cloud, as this would violate the data protection tenets in a variety of ways. For one, the SaaS provider may process or store data on servers (either primary or in backup locations) that are not physically located in a region that is covered by strict data residency laws. Next, employees of the SaaS provider have access to the data as they perform routine processes and maintenance, such as data backups or server upgrades, and these employees are frequently located in other regions.

This incidental access, though necessary under the SaaS provider's service level agreement (SLA), still violates strict policies or regulations covering who is authorized to view or handle the data. And finally, SaaS providers rarely accept full accountability in their SLAs for the security of their customers' data, leaving the customers with full liability in the event of a breach.

BLUE COAT®

Security Empowers Business

Under these circumstances, organizations have a choice to make: say "no" to the cloud and its numerous benefits, or find the means to definitively address the residency and compliance issues when working with sensitive data. Given that SaaS computing has become a multi-billion dollar business globally, it's clear that organizations are finding ways to protect their data in the cloud – some more useful than others depending on the situation and the design.

The issues discussed above largely stem from data being "in the clear."

Anyone who can access the data, whether they are authorized to do so or not, can clearly see the meaning and values of the data. The way to resolve this problem is to "obfuscate" the data – that is, make it unreadable or meaningless so that if the data is breached, it's unusable by the intruder.

There are two common data obfuscation techniques that are widely used today: encryption and tokenization.

Techniques Available to Obfuscate Information

Encryption

Encryption is the process of using an algorithmic scheme to transform plain text information into a non-readable form called ciphertext. The mathematical algorithm that turns plain text into ciphertext, or vice versa, is called a key. See Figure 1 for a simple illustration of how encryption works.

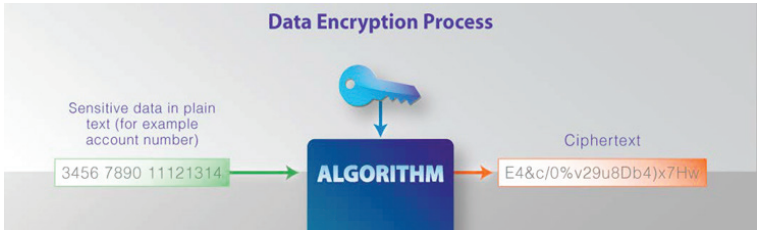

Figure 1: The encryption process

It's possible to use a single key to both encrypt the information and to decrypt (or unencrypt) it and return the information to its original plain text format. In practice, however, it's much more secure to use one key to encrypt data and another separate key to decrypt it. An even better practice is to use a unique pair of keys for each field of data being encrypted. This multiple key practice provides an extra level of security, so that if one field is compromised, the others around it aren't. See Figure 2 for a simple illustration of the use of separate keys to encrypt and decrypt information.

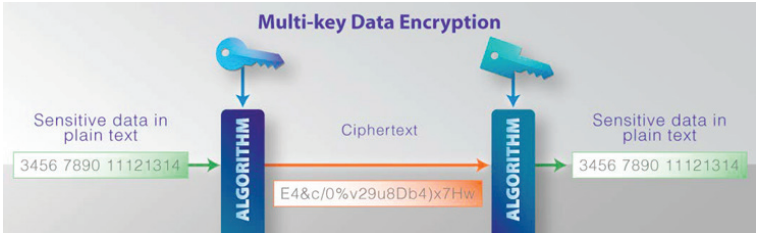

Figure 2: Encryption/Decryption using different keys

Tokenization

Tokenization is the process of randomly generating a substitute value, or token, that is used in place of real data, where the token is not computationally derived in any way, shape or form from the original data value. The most common form of tokenization uses a highly secure lookup table (called a vault) to keep track of the relationships between real data and the substitute token values. This process is illustrated in Figure 3.

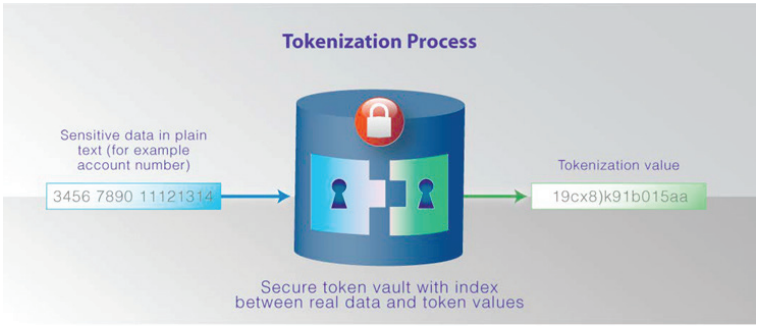

Figure 3: The data tokenization process

Because tokens are totally random, there is no relationship among them. Guessing one token doesn't get an intruder any closer to figuring out any of the other tokens. There's no key or computation that unlocks all the tokens once an intruder has figured one out. Tokenization is viewed as the strongest way to use replacement data for original clear text values. With encryption, the value of an encrypted field is reversible to get to the original value; with tokens, there's no reversibility.

The discipline of providing encryption for data in the cloud is relatively new territory. Cloud providers, encryption vendors and user organizations are trying to figure it out because it's so unlike the on premise enterprise environment where the application, data and encryption keys are all in one location.

The main challenge is in finding an effective way to cloak the data so that it isn't exposed in the clear when it is in the cloud, while simultaneously preserving the functionality of the SaaS application. The last point is critical. If the end-user experience with the SaaS application is broken, it defeats the business drivers for adopting the application in the first place. Security professionals know that they need to protect sensitive data without impacting SaaS usability. It's a conundrum that requires innovation to meet all needs.

Approaches to encryption for data in the cloud

SaaS implementations are leading to the development of various approaches to encryption. In fact, many SaaS providers are partnering with encryption vendors for their technical expertise and innovative solution architectures. Their shared mission is to satisfy industry mandates and legal requirements pertaining to protecting sensitive data in the cloud while also preserving the functionality of the SaaS application to the greatest extent possible. As described below, different encryption approaches accomplish this mission to varying degrees.

Database Level

A common tactic is to encrypt a user organization's entire database. While this approach certainly does protect the data from unauthorized users and hackers, and it satisfies many compliance requirements, it's a sub-optimal choice for SaaS applications for a variety of reasons. Even in a multi-key scenario, one key encrypts the entire table, and only one key is needed to unlock the database's entire cache of data. This process doesn't provide a granular level of restrictive access to sensitive data.

More troublesome, however, is that database level encryption breaks some of the functionality of the SaaS application when the data is encrypted. It's not possible (or it's actually meaningless) to search or sort on encrypted values. If preservation of SaaS functionality is required (and it typically is), organizations need to leave the information decrypted to allow the search or sort activity. However, this allows the application to see the complete data in the clear and in doing so would give the cloud provider access to the data, which violates key regulatory, privacy and compliance requirements.

Field-specific encryption

Field level encryption, when implemented properly, can provide all the benefits of strong encryption (e.g., FIPS 140-2 validated encryption modules), but without the downsides discussed previously on database level encryption. The user organization chooses which data fields to encrypt; not everything is sensitive, so it's not necessary to encrypt everything.

For fields that are encrypted, it's possible (and encouraged) to use a different decryption key for each field. This lets the user organization control who can access those different fields. For example, only certain employees should be able to access the social security number field. Not everyone's job role requires access to that information, so those who don't need it shouldn't be allowed to view the data in clear text.

Field-specific encryption may also improve performance; for instance, if a user can search the clear text fields on non-sensitive information without having to decrypt anything, the search can happen much more quickly.

Depending upon the implementation, even field-specific encryption can break a SaaS application's functionality if a field required for searches or sorts is encrypted. To get around this limitation, some encryption vendors modify (weaken) the encryption algorithm to preserve some transparent level of connection to the original clear-text value in order to try to keep the the application's functionality intact. But the downside here is apparent, since an organization is being asked to accept weaker data protection in order to keep the SaaS application functioning properly.

As an example, suppose we want to search on employee last name. To make things searchable, a vendor may use an encryption approach that produces some commonality in the encrypted values so the SaaS application can do the search on the backend. So, they make sure that when they encrypt words beginning with "Smi," the first three characters of the encrypted value will always be "4rl." Following that logic, when they encrypt "Smithers" and "Smithfield," the first three characters of their encrypted values will also be "4rl," as shown in Figure 5. In other words, the algorithm used to encrypt the data maintains some level of commonality among the encrypted values.

CLEAR TEXT VALUE	ENCYPTED VALUE weakend to support search
Baker	8uR4;wa
McMahon	Mv62dpy
Smith	4rl)iyr
Smithers	4rl3Mv
Smithfield	4rl[c33
Smythe	6fD06kb
Wang	$Fiqq98b
Zenovia	N?h5(gx

Figure 4: Example of encryption with a search-friendly algorithm

Maintaining this commonality in the encrypted values essentially weakens the encryption overall and makes it easier to break the encryption. If someone can figure out how to derive one field level value (i.e., get the key), they can get all the data that's encrypted with that key. Worse, they may be able to figure out the entire cryptographic process and put all the data at risk of compromise.

There's no one-size-fits-all solution for encryption in the cloud. Each company has to consider its own needs and compliance requirements. Here are some of the challenges and other factors to consider when selecting a solution.

Strength of encryption

FIPS 140-2 is the U.S. federal government encryption standard, and many businesses follow this standard as well for its strong level of encryption. Government agencies are prohibited from purchasing a data encryption module that doesn't have a FIPS 140-2 validation. An encryption vendor whose cryptographic module attains this validation attests that its solution:

1. Uses an approved algorithm,
2. Handles the keys appropriately, and
3. Always handles the data to be encrypted in a certain way, in a certain block size, with a certain amount of padding and with some amount of randomness so the ciphertext can't be searched.

FIPS 140-2 validation requires adherence to multiple levels of security that other encryption standards do not have.

For example, FIPS 197 is an algorithmic standard that addresses the Advanced Encryption Standard (AES). As a standard that is used worldwide, AES is approved by the U.S. government to satisfy only condition (1) listed above for FIPS 140-2. However, an encryption solution that only incorporates the validated algorithms of FIPS 197 does not meet security requirements (2) and (3) above, and hence is insufficient to be certified as FIPS 140-2 (minimizing its usefulness for those looking to use strong encryption).

This situation leaves the door open for confusion amid various vendor claims. Some vendors say "we can do AES encryption so our encryption is good," or, "we use Military Grade encryption." This is not a true statement if the vendor is modifying the algorithm to make sure that a few characters always line up to preserve search (as described in the earlier scenario regarding encrypting last names). This approach utilizes a weakened form of encryption that is certainly not FIPS 140-2 AES encryption. From a certification standpoint it doesn't have any strength behind it; it just has a certification that says "If you run these strings through a certain way, you will get a result that looks like this." It is important to remember that the implementation of AES without FIPS 140-2 is treated by the U.S. federal government as clear text.

Some encryption vendors need to utilize different algorithms when they are asked to preserve the SaaS application's ability to do search and sort functionality. In doing so, they are opening up their encryption engine to crypto analysis and a much easier path to cracking the data and, by definition, putting their clients at risk. User organizations have to make the difficult tradeoff between meeting requirements for data privacy/protection and the usability of their application. This is a no-win scenario.

Who controls the keys

Who controls the encryption/decryption keys is typically not a big concern when everything is done on premise within a user organization's own data center. There may be some need for separation of keys to prevent one internal person, such as a database administrator, from having unnecessary access to everything.

When applications and data move into the cloud, however, the question of who controls or even holds the encryption keys is a big concern. Allowing a SaaS provider to control one or more keys that can decrypt data – such as to enable search on the database – gives the provider access to data, which may violate regulations or internal policies the

business must operate under. The preferable stance is to ensure that the user organization is able to maintain control over the keys at all times.

Companies that are looking for a cloud encryption solution should enquire about key control when talking to encryption solution vendors.

Key management

A common security measure to guard against data compromise is key rotation, or changing the keys periodically to reduce the amount of harm that can be done if a key is compromised. But key rotation introduces its own set of challenges: How often should the keys be rotated? What should be done about data that was encrypted with an old key? Should it be decrypted and re-encrypted with a new key? Where should old keys be stored, and for how long? And of course, the more keys that are in use, the more complex the process will be for rotating them. Organizations frequently use tokenization approaches in order to avoid the complexity associated with key management.

Functionality preservation

When a user organization buys into a SaaS application, the company wants to be able to use all (or at least most) of the application's functionality. As discussed earlier, strong encryption can "break" some of the application's critical functionality because it can be extremely challenging to perform specific functions on encrypted data. We've also shown that it's possible to alter the encryption algorithm just enough ("water down" the encryption strength) to allow the use of specific functions but this has significant shortcomings. Be sure to consider this as you evaluate vendors in this space, since there are companies that do provide the ability to preserve functionality, even when strong FIPS 140-2 encryption is used.

Format preservation

Some fields may require a specific format that the encryption process must accommodate. For example, SaaS applications that store phone numbers, social security numbers, credit card account numbers, and email addresses expect the stored value to have a specific format. The application may reject data – clear text or encrypted – if it doesn't meet the format requirements. If an encryption algorithm turns "John.Doe@anyco.com" into an encrypted value that doesn't have the "@" sign or the ".com" (or .net, .edu, and so on), the application may reject the data and store a blank field, or continue to remind the user to enter a valid data point.

Data residency/privacy

Some countries or government entities specify that personal data may not be disclosed abroad if the privacy of the data cannot be guaranteed. This is often a challenge where cloud computing is concerned, as cloud providers are reluctant to guarantee that private data will always remain within the borders of the country or region. It is difficult for them operationally to support this requirement. While some Software-as-a-Service providers have taken the costly step of building out data centers in places such as Switzerland and certain countries within the European Union in order to provide an in-country or in-geography location for the data, this does not address the issues of (1) the site of the back-up data center where the information flows to in the event of an outage or (2) the fact that the SaaS provider likely has employees from other regions that need access to the customer's data for system maintenance and customer support requirements. Because of this, few SaaS providers have reached the stage where they can provide total in-country service guarantees.

Unfortunately, many enterprises have concluded that data encryption does little to satisfy data residency requirements, even when an enterprise retains the keys, since the data – albeit encrypted – is still present and can potentially traverse borders in the cloud.

Relative to encryption, tokenization is a newer technology whose use is still evolving. The most common usage so far has been in the payments space where merchants use it for PCI compliance. In this scenario, merchants store tokens instead of actual credit card numbers, also called primary account numbers (PANs), which are highly sensitive and easy to monetize if they are breached. The advantage here is that tokenization completely removes any form of the PAN from the merchant's card data environment (CDE), increasing security and reducing a merchant's PCI audit requirements while still allowing backend applications such as data analysis and marketing to function.

In the payments space, a token replaces a PAN in a one-to-one relationship. That is, the same token is always used to represent a specific card number stored with a particular merchant.

In terms of using tokens in cloud-based SaaS applications, a company with sensitive information tokenizes that information before sending only the token values into the cloud application for processing and storage. The real data is stored in the token vault, which is encrypted itself, which is on premise within the company's own data center or in another location, such as at an IaaS provider, that meets security and residency

requirements. In countries with strict residency/privacy laws, this implementation of tokenization is a well-designed approach to solving for residency requirements.

Though tokenization represents a very strong form of security, there are issues to consider with this type of obfuscation.

Functionality preservation

The main challenge with tokens is that they typically break the functionality of SaaS applications. For example, totally random token values do not support a search function. If I search on "Smit*" and a token has replaced smith in the database, the application will not know what record to pull back because there is no Smith in the SaaS data tables. Also, there's no commonality in the tokens that represent "Smith" and another record such as "Smithers" because the tokens that replaced these values were randomly generated. Of course, if the SaaS application cannot function as intended, the user organization cannot benefit from its investment in the application.

Format preservation

Tokens have the same issues as ciphertext (encryption) when it comes to format preservation, though it is possible to force tokens into specific formats to maintain compatibility with the application needs. In practice, it's actually a bit easier to preserve data formats with tokens than with ciphertext.

Which to Use: Encryption or Tokenization?

As they make plans to adopt SaaS applications, user organizations have to consider when to use each of these two obfuscation techniques. There's no clear indicator of when one solution is any better than the other. Encryption and tokenization are not necessarily competing technologies; they are often complementary to one another. In fact, encryption is an essential component of a tokenization solution because every token server relies on encryption and associated key management to safeguard data in the token vault.

An Organization Needs to Consider:

- What are the objectives of the business process using the SaaS application?
- What information needs to be secured?
- How will that data be used?
- What regulations, policies and other external requirements drive how we handle our data and do they specify a certain obfuscation approach?

Some mandates and regulations dictate that encryption must be used to safeguard the data. This may be more a factor of the newness of tokenization as an obfuscation technique than it is a statement of the effectiveness of tokenization or encryption. While encryption has existed for decades, tokenization as a viable technology has only existed since about 2005. As mentioned earlier, the most common use of tokenization today is to meet PCI DSS compliance requirements. However, as of the writing of this document the PCI DSS specifications do not say anything at all about tokenization. (The PCI Security Standards Council is only now considering tokenization as an added security measure.) That said, a user organization must still consider if it is under a mandate to use encryption as a specific security solution.

Encryption has advantages over tokenization in certain circumstances. As discussed earlier, organizations willing to accept less stringent obfuscation approaches can weaken encryption algorithms in order to support specific functionality in a SaaS application. (Note that this is not permissible for U.S. federal government agencies or their contractors that must adhere to the FIPS 140-2 standard.) Tokenization solutions also require more storage capacity than encryption solutions, so this needs to be taken into consideration for use with large databases.

Similarly, tokenization has its advantages over encryption in certain circumstances. It is most helpful in meeting residency/privacy requirements which prohibit data – in the clear or as ciphertext – from crossing a physical border. User organizations that must keep sensitive data within a specific country or region can do so by securing the real data in a token vault on premise while placing meaningless tokens in the cloud. What's more, for common SaaS data storage and data validation, tokens are generally better than encryption at maintaining format requirements such as field lengths and character validation.

In addition, there are operational factors to consider. Encryption requires key management, rotation, custodians, policies, and so on. Tokenization requires the careful management of the token database (and the infrastructure to support a token database). Many organizations find that tokenization is less taxing to manage from an IT perspective, but others have automated key controls and existing policies and tools that facilitate the use of encryption.

An organization can use a hybrid approach, using both tokenization and encryption to safeguard data; it doesn't have to be an either/or proposition. Given the right security platform, the organization may be able to use one product that can tokenize one field or fields and encrypt others based on specific business needs.

Cloud computing is here to stay as a cost effective alternative to on-premise computing. In particular, companies want to take advantage of a multitude of SaaS cloud offerings because the benefits are too great to ignore.

What's Needed to Support Encryption and Tokenization in a SaaS Public Cloud Environment

The need for data security, compliance and residency is driving companies to look for encryption and/or tokenization solutions to support their SaaS initiatives. Rather than select point solutions to serve each individual application, it's better to address the enterprise needs as a whole. Here's what's needed to support encryption and tokenization in a SaaS public cloud environment.

Strong security

An enterprise solution must provide strong FIPS 140-2 encryption (where the enterprise owns the keys) and/or strong tokenization – preferably both – or it really isn't a good solution at all. While "weakened" solutions may offer interesting ways to preserve application functionality – for example, allowing a few characters to stay in the clear to enable search functions – such solutions simply reduce the overall security posture and lead to vulnerabilities and risk. This, of course, defeats the purpose of applying encryption or tokenization to sensitive information in the first place.

Functionality preservation

The primary reason companies deploy SaaS applications is to increase business productivity. Saving money and reducing IT overhead are also good reasons, but if the application can't help workers become more productive in their jobs, there's little sense in deploying the application. This is why it's so important to preserve the functionality inherent to the application. If workers aren't able to perform critical tasks, such as search and select data or print meaningful reports, they will reject the application as not meeting their needs.

The conundrum is that the need to support an application's functionality is in conflict with the requirement to maintain strong security. User organizations often feel they are forced to compromise in one area or the other, but they can find vendors that will enable them to meet both sets of needs.

Flexibility

An enterprise solution must be built to provide flexibility and choice. Ideally, the solution would support both encryption and tokenization and allow the user organization to choose which obfuscation technique to use when, and how, even on a field by field basis. For example, a company might choose to tokenize fields with personally identifiable information (PII) to meet residency requirements, and to encrypt account descriptions for security reasons. The organization also should have the ability to make changes over time as their circumstances change. Data fields that they tokenize today may be more appropriate to encrypt in the future based on updates in regulations. Having the ability to easily make these sorts of changes is critical.

Robust platform

Enterprise security should be holistic. Therefore, an encryption/tokenization solution should be architected as a platform that can support multiple SaaS clouds and allow for hybrid cloud implementations. That is, the user organization should be able to configure the solution to perform functions such as data processing, management and storage locally if desired, while maintaining encrypted and/or tokenized data in the cloud instances. The platform needs to also allow for varying deployment options based on the organization's security principals and operational requirements. For example, it may want certain elements of the platform deployed in a DMZ, some elements behind the corporate firewall, some in an Infrastructure-as-a-Service (IaaS) partner, etc.

Case Studies

The following case studies illustrate the flexibility and power of an enterprise security solution that provides both encryption and tokenization functions. In all cases, the organizations are using the Blue Coat Cloud Data Protection Gateway to maximize their capabilities while ensuring a secure data environment.

Data residency

An international bank in Europe has customers in both Switzerland and Germany. When implementing a SaaS application in the cloud, the bank was challenged by the requirement to meet two distinct and very strict laws on data residency and privacy.

For the accounts pertaining to German customers, the bank would have to adhere to Germany's Federal Data Protection Act and the European Data Protection Directive of the EU. And for the accounts belonging to Swiss customers, the bank was under mandate of the Swiss Federal Act on Data Protection.

Each set of laws require that the personal information pertaining to specific customers remain within the physical borders of the customers' home countries. In essence, German customer account data has to remain in Germany/the EU and Swiss customer account data has to remain in Switzerland. What's more, only German employees are permitted to view the German customer accounts, and only Swiss employees are permitted to view the Swiss customer accounts.

The SaaS provider didn't have the means to split the data among the two locations and assure the privacy of the accounts as dictated by the laws. The architecture of the Cloud Data Protection Gateway allowed Blue Coat to devise a solution that uses the IP addresses of the bank employees to determine whose data they could view. Blue Coat also enabled the bank to house the clear text data in a token database that is resident in each customer's home country – one token vault in Germany and one vault in Switzerland. Tokenized data is then sent to the single cloud-based SaaS application.

The solution is sophisticated and elegant at the same time. More importantly, it enables the bank to use its preferred SaaS application in the cloud while adhering to each of the countries' data residency/privacy laws.

Third party obligations

A public sector organization in the United Kingdom is performing services on behalf of the UK government. As a condition of the business contract, the organization is required to observe the same regulations as the government agency it serves. The business process requires the company to handle citizen data, which under law is not allowed to leave the UK.

The company wanted to use a cloud based SaaS CRM application from a U.S.-based company. Due to concerns about data privacy, the company implemented a true hybrid cloud where the UK citizen data and the Blue Coat infrastructure are housed in the UK. The SaaS application is hosted in the U.S., but all the sensitive information handled by the company is actually stored in a UK data center. Tokenized data has replaced the citizen data in the SaaS application, thus fulfilling the public sector organization's commitment to protect the privacy of the data by ensuring it stays resident in the UK.

Regulatory compliance

Healthcare providers need to ensure that Protected Healthcare Information (PHI) gets processed and stored in certain ways based upon rules specified in the Health Insurance Portability and Accountability Act (HIPAA). A healthcare provider wanted to move to the cloud for their new CRM system, but concerns regarding the security of putting PHI in the cloud were delaying the project. Encrypting the PHI would be advisable per HIPAA guidelines, but the business users were concerned about usability implications on the system.

The organization is able to strongly encrypt all sensitive PHI stored or processed in the CRM system by using the Cloud Data Protection Gateway. The end-user experience of the CRM application is not altered and the organization can satisfy the best practice of securing information via encryption techniques.

Conclusion

Cloud-based SaaS applications are only growing more popular with customers in virtually every industry and every global region. It's rare that a user organization would not be under some constraint that mandates the strict protection of sensitive data. Sometimes it requires real innovation to build an obfuscation solution that complements the SaaS application so well that the user organization can both reap the benefits of SaaS and adhere to all required specifications. It can be done, and there's no need for organizations to say "no" to the cloud and its many benefits.

BLUE COAT®

Security Empowers Business

v.WP-DATA-PRIVACY-AND-COMPLIANCE-IN-THE-CLOUD-EN-v1b-1015

Blue Coat Systems Inc.
www.bluecoat.com

Corporate Headquarters
Sunnyvale, CA
+1.408.220.2200

EMEA Headquarters
Hampshire, UK
+44.1252.554600

APAC Headquarters
Singapore
+65.6826.7000