Security Issues in Cloud based e-Governance model

Rajan Manro  Dr. Rajneesh Randhawa  Dr. A S Joshi
Asst. Professor & Coordinator  Asst. Professor  Former Director
DBIMCS  Punjabi University Patiala  PIMT, Mandi Gobindgarh
Mandi Gobindgarh

ABSTRACT

Government Bodies and IT departments of any state in India can make considerable savings switching from in-house e-Governance systems to cloud-based e-Governance solutions using software-as-a-service (SaaS) model of cloud computing. Large government offices should consider using the cloud model for e-Governance services delivery.

Just like a traditional e-Governance software solution, Cloud based e-Governance must provide physical security, transmission security, storage security, access security, data security, and application security. In this paper, we will try to focus on these broad classifications for discussing the similarities and differences between Cloud e-Governance and traditional e-Governance software security.

Keywords
Cloud, e-Governance, SaaS, physical security, transmission security, storage security, access security

1. INTRODUCTION

In today’s highly competitive world, large or middle business entities are challenged by rising material costs, increasing competition, and ever-changing customer requirements. To thrive, they need to stay agile, maximize productivity and ensure their systems actively assist them to work smarter. e-Governance based solution should be up to par, to control these factors like dizzying levels of complexity, duplication of effort, and in the worse cases, poor quality and customer service. Cloud based e-Governance can be a better solution but security issues are always there. This paper will analyze some of the issues with traditional model of e-Governance.

2. PHYSICAL SECURITY

In a cloud application data must be located somewhere. The physical surroundings of the software and data is an important component of a government process plan as well as a software security plan. A physical security breach means that somebody with malicious intent has physical access to the hardware where either your application is running or where your data is stored. If other forms of security are in place, a physical security breach will not result in loss of data. However, if the intruder’s intent is to disrupt your service, then a lapse in physical security will be a problem. Part of any e-Governance service plan should include a solid physical security plan.

Cloud Differences: Applications running in an external cloud require passing data between the cloud and the user location. Frequently this occurs over the Internet and over wireless networks. Furthermore, client machines are mobile (access from anywhere being a big advantage of the cloud) so processing power and bandwidth may be at a premium. Web-based systems utilize a browser on the client device and take advantage of SSL encryption to protect all communications with the server. The SSL algorithm is supported by all major browsers and encapsulates application-specific protocols like HTTP to form HTTPS so no one can hijack a session or read the data. SSL requires negligible computing overhead and is acceptable security for banking, health care, and other sensitive industries.

Some folks ask about SOAP and how that differs from HTTPS. HTTPS helps you communicate between the server, and the database, there is a chance that transmissions can be intercepted. An easy way to prevent this involves encrypting all communications between source and destination. However, encryption comes at a cost to performance. If you spend too many processing cycles encrypting and decrypting data, you will have to purchase more expensive hardware or endure delays.

There are several types of security algorithms that are used to protect communications. The underlying idea is that sensitive or private data is scrambled using an encryption key and a data encryption algorithm. The data cannot be read or deciphered without the decryption key. The decryption key can be the same (symmetric) or different (asymmetric) from the encryption key. Once scrambled, the data is sent to its destination. If intercepted, the data can only be reconstructed by using an algorithm that tries to guess the description key – a process that takes many years using powerful computers. When the scrambled data arrives at its destination, the receiving party knows the proper decryption key by querying a key master or certificate authority.

Cloud Differences: Applications running in an external cloud require passing data between the cloud and the user location. Frequently this occurs over the Internet and over wireless networks. Furthermore, client machines are mobile (access from anywhere being a big advantage of the cloud) so processing power and bandwidth may be at a premium. Web-based systems utilize a browser on the client device and take advantage of SSL encryption to protect all communications with the server. The SSL algorithm is supported by all major browsers and encapsulates application-specific protocols like HTTP to form HTTPS so no one can hijack a session or read the data. SSL requires negligible computing overhead and is acceptable security for banking, health care, and other sensitive industries.

Some folks ask about SOAP and how that differs from HTTPS. HTTPS helps you communicate between browsers and servers, but SOAP provides secure communications between applications. SOAP encapsulates additional data in the form of XML so cloud applications can communicate more efficiently than if they were required to send a series HTTP requests.

technology (finger print, ID badge, retina scans) are often implemented.

SaaS applications are run by administrators who are employed by the software vendor or cloud provider and not the company who purchased the e-Governance software. The quality and reliability of administrators depends more on the resources and focus than the employer.

2. TRANSMISSION SECURITY

When data is communicated between the user, the server, and the database, there is a chance that transmissions can be intercepted. An easy way to prevent this involves encrypting all communications between source and destination. However, encryption comes at a cost to performance. If you spend too many processing cycles encrypting and decrypting data, you will have to purchase more expensive hardware or endure delays.

There are several types of security algorithms that are used to protect communications. The underlying idea is that sensitive or private data is scrambled using an encryption key and a data encryption algorithm. The data cannot be read or deciphered without the decryption key. The decryption key can be the same (symmetric) or different (asymmetric) from the encryption key. Once scrambled, the data is sent to its destination. If intercepted, the data can only be reconstructed by using an algorithm that tries to guess the description key – a process that takes many years using powerful computers. When the scrambled data arrives at its destination, the receiving party knows the proper decryption key by querying a key master or certificate authority.

Cloud Differences: Applications running in an external cloud require passing data between the cloud and the user location. Frequently this occurs over the Internet and over wireless networks. Furthermore, client machines are mobile (access from anywhere being a big advantage of the cloud) so processing power and bandwidth may be at a premium. Web-based systems utilize a browser on the client device and take advantage of SSL encryption to protect all communications with the server. The SSL algorithm is supported by all major browsers and encapsulates application-specific protocols like HTTP to form HTTPS so no one can hijack a session or read the data. SSL requires negligible computing overhead and is acceptable security for banking, health care, and other sensitive industries.

Some folks ask about SOAP and how that differs from HTTPS. HTTPS helps you communicate between browsers and servers, but SOAP provides secure communications between applications. SOAP encapsulates additional data in the form of XML so cloud applications can communicate more efficiently than if they were required to send a series HTTP requests.

technology (finger print, ID badge, retina scans) are often implemented.

SaaS applications are run by administrators who are employed by the software vendor or cloud provider and not the company who purchased the e-Governance software. The quality and reliability of administrators depends more on the resources and focus than the employer.

2. TRANSMISSION SECURITY

When data is communicated between the user, the server, and the database, there is a chance that transmissions can be intercepted. An easy way to prevent this involves encrypting all communications between source and destination. However, encryption comes at a cost to performance. If you spend too many processing cycles encrypting and decrypting data, you will have to purchase more expensive hardware or endure delays.

There are several types of security algorithms that are used to protect communications. The underlying idea is that sensitive or private data is scrambled using an encryption key and a data encryption algorithm. The data cannot be read or deciphered without the decryption key. The decryption key can be the same (symmetric) or different (asymmetric) from the encryption key. Once scrambled, the data is sent to its destination. If intercepted, the data can only be reconstructed by using an algorithm that tries to guess the description key – a process that takes many years using powerful computers. When the scrambled data arrives at its destination, the receiving party knows the proper decryption key by querying a key master or certificate authority.

Cloud Differences: Applications running in an external cloud require passing data between the cloud and the user location. Frequently this occurs over the Internet and over wireless networks. Furthermore, client machines are mobile (access from anywhere being a big advantage of the cloud) so processing power and bandwidth may be at a premium. Web-based systems utilize a browser on the client device and take advantage of SSL encryption to protect all communications with the server. The SSL algorithm is supported by all major browsers and encapsulates application-specific protocols like HTTP to form HTTPS so no one can hijack a session or read the data. SSL requires negligible computing overhead and is acceptable security for banking, health care, and other sensitive industries.

Some folks ask about SOAP and how that differs from HTTPS. HTTPS helps you communicate between browsers and servers, but SOAP provides secure communications between applications. SOAP encapsulates additional data in the form of XML so cloud applications can communicate more efficiently than if they were required to send a series HTTP requests.
3. STORAGE SECURITY

When e-Governance data is accessed by users, business logic limits unauthorized access to users with the proper credentials (see section on application security). But suppose a network administrator has access directly to data in the database. In this case, the data could be viewed without going through the business logic.

To protect against this vulnerability, sensitive data should be encrypted when it rests in the database or in a file system. This prevents direct access and ensures that all data is only accessed via the application logic. The application knows how to decrypt the data, so a legitimate user will not be impacted. As with transmission security, the encryption and decryption processes create processing overhead, so non-sensitive data should be stored in the clear to minimize costs. Additionally, make sure that any required data indexing is not broken in the encryption process.

**Cloud Differences:** In cloud systems, data is stored in a remote location on servers maintained by a cloud provider. The cloud provider *should* have procedures in place to ensure that there is no direct snooping into client data. But somebody has to be responsible for database administration, and usually this person is not employed by the client. The ability to pick and choose fields to encrypt on the database is important to provide protection without adversely impacting performance.

4. ACCESS SECURITY

Access (or perimeter) security is important for preventing unwanted users from grabbing resources and sending unauthorized queries to your servers. Usually this is accomplished through the use of firewalls that prevent unwanted traffic from communicating with your business applications. Lack of access security could impact your application availability (in the case of a denial of service attack) and provide hackers with a way in to make it easier to steal resources or passwords.

There are many types of firewalls, network level firewalls (fast inspection of IP, port, and service in the packet headers), circuit level firewalls (monitor sessions between computers), application level firewalls (inspect data content to protect against viruses and intruders), network address translation devices (NAT — assigns private IP addresses that cannot be reached from outside the network), and proxy servers (application level firewall that mediates transactions between computers). Network and circuit level firewalls can be implemented in an appliance or as software. Application level firewalls are most frequently implemented as software to allow for specific configuration requirements.

Additional details of perimeter security devices are well documented elsewhere.

**Cloud Differences:** Cloud systems should be protected by perimeter security — just as you would protect any on-premise application. Verify that your cloud provider has firewall protection in place to prevent intruders and denial of service attacks. Multi-tenant cloud applications is slightly different because by definition, multiple users are accessing the same application code and the same resources. In this case, processes must be in place to ensure that bad things do not happen to customer A if customer B’s application is compromised.

5. DATA SECURITY

Data security limits access to data objects to specific individuals. Different levels of data security include read-only, edit, insert, and delete. Data security can be set at the application or object level.

Data security for e-Governance systems may be enforced through business logic or at the database layer. In most cases the business logic authenticates users and provides them with specific rights to data objects. This means that authenticated users gain access to objects based on specific capabilities assigned by the system. For example, a person may have read-only access to government-related information so he cannot change the values associated with the application. A person or user may have access to citizen records that he manages, but not have access to citizen records managed by others. To simplify e-Governance, systems offer role-based security so administrators can assign broad security policies to specific individuals. By assigning roles, administrators can change security for many people at once without the responsibility of changing individual records.

Most data security is limited to data access. Once a user gains access to specific information, screens, or reports, the information can be downloaded and shared with others. Digital rights management goes one step farther by “wrapping” data objects with rights that follow the object no matter where it goes. In this case, users can forward the encrypted data, but that data cannot be viewed or changed unless the recipient can be verified.

**Cloud Differences:** Data security in cloud applications is similar to traditional applications. Once individuals gain access to the system, the government business logic controls the specific capabilities that individual users can perform on different objects. In some types of multi-tenant SaaS applications, database level security may be utilized as an additional measure to separate data objects from different companies.

6. APPLICATION SECURITY

Application security encompasses two major areas — the way the application authenticates and manages users and the way in which application code is managed.

6.1 User Authentication

User authentication usually involves username and password to identify legitimate users. User identity is critical not only for establishing data rights, but also for creating an audit trail of activities for compliance purposes. Modern systems require strong passwords, enforce lock-out from excessive failures, and give administrators the option to require users to change their password on specific time intervals. In addition to these common security measures, administrators may restrict access to the system by IP address to combat hackers that try to guess usernames and passwords from remote locations.

Borrowed or stolen passwords can circumvent the most sophisticated technologies, so administrators may require two-factor authentication through the use of security tokens. Augmenting passwords with key fobs that generate time limited passwords provides physical access control from anywhere at a very reasonable cost. Secondary, one-time
passwords can also be sent to a mobile phone via SMS. Products include Secure ID from RSA and many others.

Authenticated users are granted access to specific data and processes. The e-Governance application must provide security measures to prevent authenticated users from doing unauthorized tasks. For example, somebody authorized to input data should not be able to delete data. If somebody is authorized to fill out a form, the data must be examined to prevent overflow and SQL injection issues.

6.2 Managing Code and Logic

All e-Governance software undergoes revisions and updates. The processes that manage these updates can be included as part of the overall security plan provided by the vendor. For example, when compiling the final code, processes are in place to insure that rogue code is not inserted into a production build.

Cloud Differences: A major benefit of the cloud is the ability to run your business from anywhere on any browser. In client-server environments that sit behind firewalls, clients are often “trusted” machines. To equal this level of security, cloud systems require all business logic to run on the server and consider all clients as untrusted. By enforcing all business logic at the server, cloud systems are actually more secure than client-server systems that rely on files and components installed on user machines.

7. SUMMARY AND CONCLUSION

Cloud e-Governance and traditional e-Governance share many of the same security issues. In the case of traditional e-Governance, security issues are managed by internal IT resources. In the case of Cloud e-Governance, similar issues are managed by external resources. A third party audit can be done to certify that external processes are documented and followed – but in most cases, the vendors pay more attention to these details than internal resources would.

In a cloud deployment with shared hardware, a shared operating system, and customer-specific application code, the security issues are almost identical to traditional e-Governance. Distances covered by transmission security are longer, but that has little impact on overall security. When the cloud is running a multi-tenant application, the data security and application issues are slightly different, but not necessarily less secure. In a multi-tenant deployment, the application must be designed to prevent client 1 seeing client 2’s data. As far as I know, all Cloud applications are designed in this way. The multi-tenant application must also allocate resources so client 1 cannot steal resources from client 2 during a period of heavy usage.

When using a web-based cloud application, client software is replaced by a browser. From a security perspective this usually has little impact because in a well written cloud application, all application security is performed on the server. The client software is inherently not trusted.

8. REFERENCES


