Security Case Study:
Eurocenter Maximus Information Platform

HASITH D. YAGGAHAVITA
INFORMATION SECURITY CONCEPTS AND PRINCIPLES
M.SC. IN ENTERPRISE APPLICATION DEVELOPMENT
12TH JULY 2008

Table of Content

Introduction

This case study assesses the security implementation of 'Maximus information platform', one of the products offered by Eurocenter DDC [1] Sri Lanka.

In the first part of the report, Maximus information platform is introduced with discussing the introductory aspects such as business case, technical implementation overview and security requirements. Then the case study discusses various security implementations of Maximus in much detail. This discussion is based on the four main security principles namely isolation, diversity, redundancy and fail-safe.

In the final part of the report, security implementation of the Maximus system is evaluated in respect of the overall security viewpoint. The key security concerns are summarized and possible improvements are discussed for better securing similar type of systems under this section.

System Background

Business Case

Eurocenter 'Maximus information platform' [2] is a decision support system that enables access and analysis of enterprise information. This platform is completely designed and developed by a team in Eurocenter DDC Sri Lanka and used by several product vendors in Scandinavian region. Maximus system consists of three major subsystems namely web portal subsystem, reporting subsystem and search subsystem.

Features such as data searching, discovery, dimension analysis and relation analysis are available through the search subsystem of Maximus.

Figure 1: Some screens from Maximus search subsystem

Reporting subsystem allow user to configure and obtain various reports as required. The data selection and report formats can be specified by the end users before they run the report. Additionally any advanced user may extend the available report types by introducing new reports with appropriate data fetch queries.

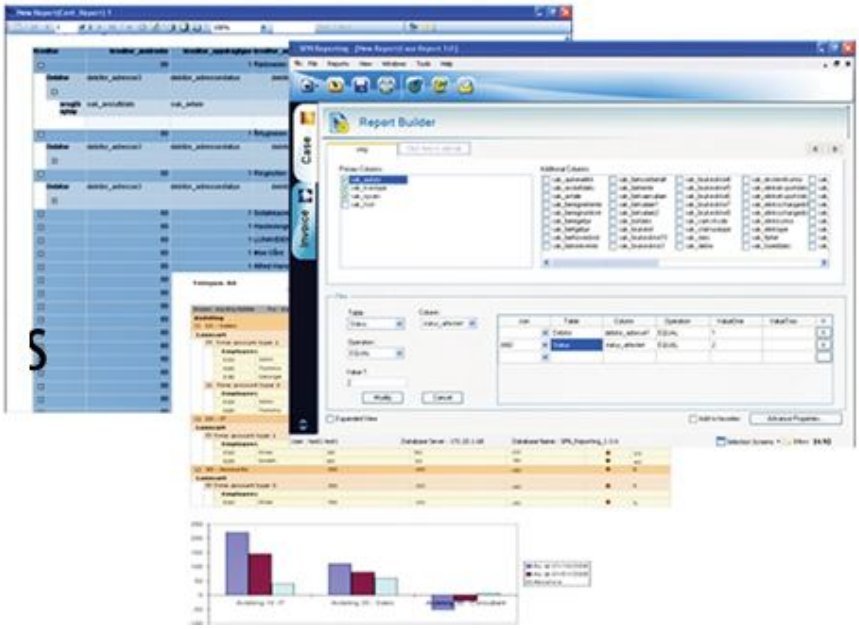

Figure 2: Some screens from Maximus reporting subsystem

The portal sub system allows users to build their own personalized dashboard that consists of summary information of reports and search analysis that are to the interest of the user.

Figure 3: Some screens from Maximus portal subsystem

The target market of the Maximus platform is ISVs who own products. Maximus can be integrated to these products and configure to make use of the data available in the product. Maximus platform is built with an extensible and well defined architecture that allows close integration to the products developed by ISVs.

Technology Overview

The platform operates on top of the existing data sources of the product to which it is integrated. Maximus supports any database system as its data source and capable of integrating data from several different databases.

Main technical modules of the platform are depicted in the following diagram.

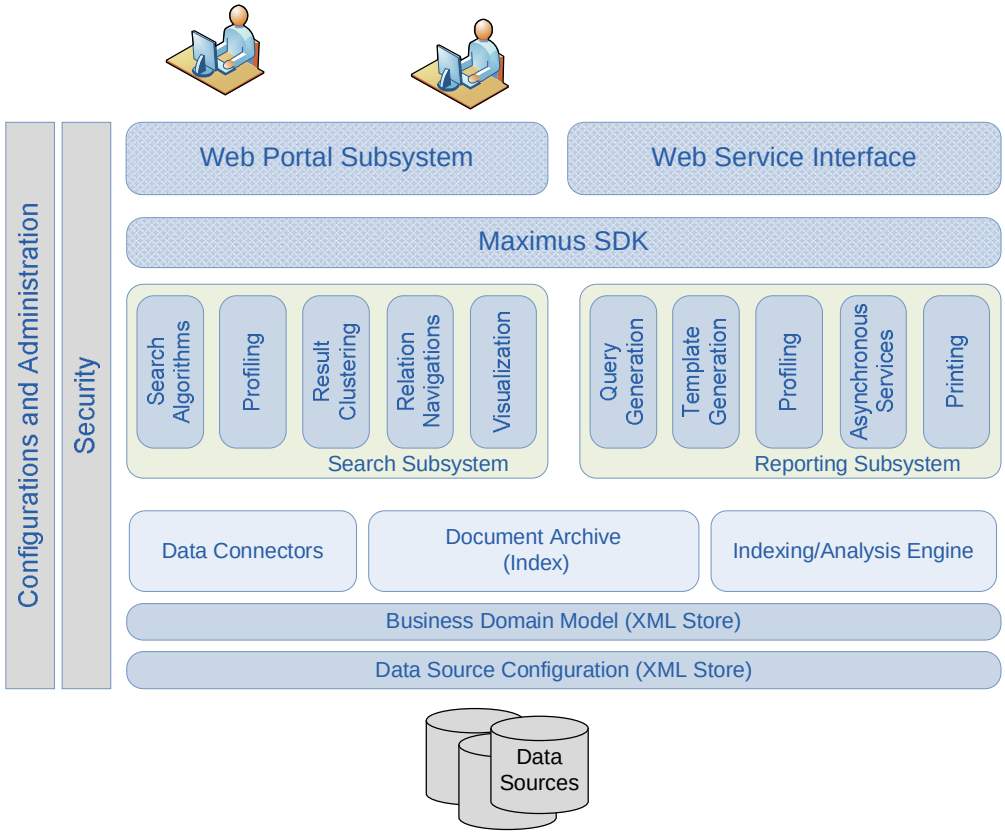

Figure 4: Maximus architecture overview

System can be deployed as a centrally hosted ASP server [3] that can host multiple instances that are used by multiple customers. The data of these customers is used to build high performance MOLAP indexes [7] on which the querying happens. Maximus is completely built on Microsoft technologies using Microsoft .NET C# language.

Security Requirements

Maximus platform deals with the most critical asset of any organization, which is the central data store. Also a single deployment of the platform may serve several different customer organizations under ASP model described above. The isolation of data becomes very critical for data security in such scenarios.

Maximus services are used by employees of all the levels in a particular organization. It should be made sure that the data is properly authorized according to the access rights of the users. Securing enterprise data from information theft or tampering is important especially as the system get exposed over the internet in a typical deployment scenario.

On the other hand, as the Maximus product can get deployed in several different locations, it should be assured that the same process followed in breaking in to a particular deployment can not be used to compromise all other deployments everywhere else. Diversified implementations are often preferred to reduce the risk of correlated failure propagation of this nature.

With the fact that Maximus system deals with most valuable asset of the organization, it is of high importance to ensure that the maximum security is enforced to the critical parts of the system. Use of redundancy to increase survivability of security systems is highly favored in Maximus security design.

Further it is important that if there is a failure in the system, the failed behavior should not cause the system to behave in an insecure manner. For an example, if access mechanism for one user is compromised, it should not compromise access information of any other user.

Security through Isolation

Network Isolation

Maximus platform uses isolation at several levels to secure itself. At the network level the network is isolated depending on the network trust levels. Properly configured firewalls [5] are used to open gates between the zones if required.

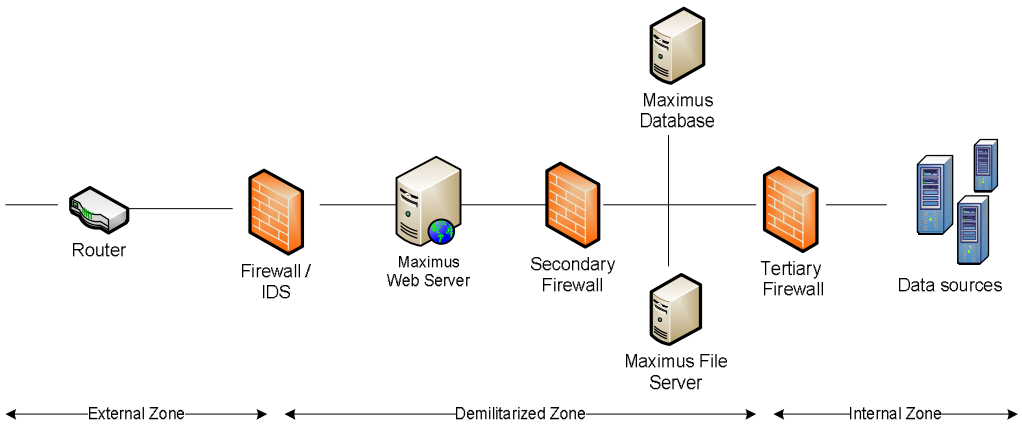

Figure 5: Maximus deployment configuration

Above diagram shows a configuration of simple Maximus deployment option. External zone directly exposed to the internet and contain no servers but network devices and security systems. Any unintended traffic is filtered by the firewall and IDS [6] deployed in this zone.

The internet servers are in DMZ [4] which has limited access to the internal data sources. For example the ports that are used by database servers for its internal purposes are not accessible from DMZ. This ensures any intruders that gain access to DMZ, still has limited control over the data sources. Secondary firewall isolates the Maximus web server and with file/database servers. Further this essentially ensures that these servers are in different physical machines creating more isolation.

Internal zone consist of the most valuable data assets and hence has more restricted access from outside. Maximus system only requires read access to the data sources in the internal zone. Therefore the user accounts used by Maximus platform only consist of read rights on the data sources. This authorization enforces the access rights isolation at system level and therefore no data tampering is possible through compromised demilitarized zone. Similarly only the essential access rights are granted on the shared file server directories.

Memory Isolation

The Application Domain [8] concept of Microsoft .NET framework provides sandboxes for .NET applications. An AppDomain is a container/secure boundary, for code and data used by the .NET runtime. The goal of an AppDomain is to isolate the applications within it from all other application running in other domains. That is, applications are protected from being affected by other applications running in different application domains.

Maximus platform is deployed on Windows IIS server [10]. Single IIS server can contain several web applications, but to avoid memory access violations of one application affecting the Maximus platform, IIS assigns a separate app-domain for Maximus. This memory isolation ensures the Maximus is least affected by physical machine memory compromise from an intruder.

Account Isolation

As specified above Maximus platform is deployed on IIS web server. The web server executes the application with permission of a system account having only the essential access rights. For example, any file system access other than the intended file shares is restricted for the system account running the application.

In the user account perspective, Maximus platform is used by several different user groups having different levels of trust levels. Most powerful of the users are administrators who can access and manipulate configuration information. Maximus has built in restrictions for isolating the trust levels of these user groups. For example, the administration of the Maximus system is only allowed for users accessing from internal zone. Similarly authentication mechanism is built around the principle of user roles and role isolation.

Communication Isolation

Various client applications are allowed to communicate to the Maximum server for business operations. Privacy and integrity is the most important security demands to be met. The encrypted messages create isolated tunnels between the parties to ensure security. Communication channel isolation serves as the primary mechanism of protecting data on the wire.

Perimeter Isolation

Each physical machine of a Maximus deployment has a defined responsibility associated with it. The principle of ‘fail-safe defaults’ is followed by opening the absolutely required entry points to/from the physical machines to the outside world. Every machine need to under go a hardening test (vulnerability scanning) before being deployed in a production environment to ensure safety.

Security through Diversity

Network Diversity

In the Maximus security specification, it is recommended that the security devices deployed in the network to be from different vendors. For example, three firewalls from different vendors can maximize the security via diversity. Realized vulnerability of one firewall may not easily be used to compromise the others when they are different from each other. For the same reason, recommendation is to select operating systems of the servers to be different Windows server versions. Maximus specification proposes the most effective mix of the servers for this matter.

Technology Diversity

During the design of Maximus system, it was decided to use dependent components from several vendors rather than using components from a single vendor. One of the reasons behind the decision is to maximize the diversity.

For example, Maximus framework uses much of the infrastructure from Microsoft and therefore several components (such as Office Document handling components) from different vendors were preferred over Microsoft built-in components.

Data Source Diversity

The connection mechanisms to the data sources are made pluggable by the Maximus architecture. This allows different connection mechanisms to be used for connecting different data sources. This diversity makes correlated attacks on different data sources much harder for an intruder.

On the other hand the Maximus system demands defining alias for all tables and columns in a database. Use of different naming other than actual physical naming hides the important information about the connected data sources.

Version Diversity

Maximus development team changes implementation of some high important critical components time-to-time on different versions they release. This is often planned with security improvements and new version availability of dependent systems such as database and application servers.

Also different implementations for different customers carry considerable customizations that are inline with the security requirements of the customer. This version diversification also makes correlated failures much difficult for an intruder.

Time Diversity

Maximus platform uses the concept of authentication tokens that are used to authenticate a client call among different components (single-sign-on). This token carries a special ticket that carries a valid hash to represent the identity. The ticket expires after a certain time period. Due to the diversity of the ticket, even if the ticket is captured by an intruder it will not be valid for much time.

Security through Redundancy

Redundant Validations

Often input data validation is performed at the client side in the user interfaces of Maximus system. But just having client side validations are not sufficient as an intruder may bypass the client layer and attack the server interfaces directly. Therefore all the validations are redundantly performed in the server as well. Further the Maximus exposes a SDK layer that can be used by other applications and validations are performed at that layer too.

Redundant Devices

In reality, security measures taken can not totally protect a device against all known and to-be-discovered attacks and vulnerabilities. Therefore, it is important to have a suitable redundancy mechanism in place that allows a backup device to take over the functionality if primary device is compromised.

For example as depicted in the figure 5, redundant firewalls are deployed in front of the valuable data assets of the Maximus system. This ensures that an intruder needs to break through several layers of firewalls before gaining the access to data sources.

Redundant Algorithms

Maximus platform supports wide variety of security algorithms in supporting various client applications. For example TLS implementation for Maximus services should have redundant cryptographic methods that can be applied for different clients as required. This makes sure that limitations in the client devices minimally affect the operation of the system.

Redundant Audits

It is important to employ audits at several levels to monitor any suspicious activities around the system. For example in a Maximus deployment, the auditing happens at the firewall, IDS, internally through the system as well as at the data source levels. The audit records are continuously monitored to identify any malicious attempts through log consolidation mechanism.

Security through Failure Strategy

Persistent Secure State

Maximus system ensures the possible failures do not put the system in to an insecure state. That is any failure should not undo the security measures that are already in place at the point of failure. For example the secure connectivity failure will not bring the system down to plain text connectivity but will abandon the connection attempt.

Backdoor Prevention

Also any nontrivial unexpected error in the system will make sure the current authentication tickets become invalid for further use. That is if the system is not certain about the operational conditions after the error, it should prevent further operations coming from the failed context.

Information Protection

Maximus platform implementation has taken measures to make sure the failures do not lead to any sensitive information exposure. For example, any error trace should not be exposed to the outside at any failure of the system. Maximus system immediately switches to a sandboxed error mode once an error is identified.

Discussion and Evaluation

In the previous sections we have discussed the security measures taken by the Maximus information platform in detail. As we have seen it is evidenced that the designers of Maximus system have taken a conscious attempt to build a firm security infrastructure around the system. As we can see most of the technical aspects of security are considered in implementing the Maximus system.

As we will see shortly the attention to the less-technical security aspects is not much evidenced in the system documentation. This is a common mistake we see in most of the system designs today.

Soft Security Measures

The focus on the soft aspects of security such as security policies and human factor is not documented in the Maximus system implementation. With the advent of complex security mechanisms, the human factor becomes the weakest link of the system security chain. Maximus system needs to have policies and guidelines defined in order to prevent security breeches.

Educating the users and accountability delegation are essential to maintain the level of security demanded by the high sensitivity of Maximus data assets. This additional layer of soft security measures essentially narrow down the window of possible security attacks.

Recovery Measures

It is important to understand that perfect security implementation is not practical in any sense in an information system. Therefore Maximus system should define recovery strategies to face emergency situations arise in breach of security.

Unfortunately planning for disaster recovery and contingency is also not much visible in the Maximus system documentation. Maximus system should ensure the valuable assets of the system can survive even after an attack on the system.

For example the data assets should be regularly backed up so that the system restore is possible from the last restore point. In addition the transaction from the last backup have to be transaction logged, making it is possible to bring the system to the original state as it was at the time of attack.

Development Environment

It is important to understand that a breach of security is not happening only in production phase of the system. It is very much possible that an intruder gain access to the development environment and exploit the system codebase to fulfill the needs. Maximus team should certify that the development environments are properly protected with appropriate policies to avoid such circumstances. Adhering and obtaining a certificate such as ISO 27001 [9] can assure the necessary trust level required for the development environment.

Conclusion

In today's context effectively achieving security needs balanced focus on all the aspects of security principles. It is important that the security designers identifying the pulse of possible attacker mindsets.

But unfortunately as discussed above we often see the systems not having much focus on the soft aspects of security such as weak human factor. But as commonly stated a system is secure as much as the security of its weakest link.

On the other hand the each and every techniques considered in isolation will not provide the intended security. Most of the security principle are correlated to each other and affected by each other.

For example the redundancy without diversity may not be useful in most of the situations. If redundant firewalls at two different levels are identical, the hacking method used on the first layer may be easily employed to break the secondary layer too.

Similarly redundancy can have conflicts with isolation principles. For example the parallel redundant firewalls placed essentially increases the attack surface to an intruder and hence can conflict with the interests of isolation requirements of the system.

In overall, the use of several different measures in several dimensions adds complexity to the total system. Unfortunately the complexity is one of the factors that increase the vulnerability possibilities of a system. More complex the system is, it becomes more complex to monitor and secure. It very important that we do not over complicate the system with the measures we introduce.

As we have seen, security implementation for a particular system needs evaluating the options in several different perspectives. Security implementations should not collide with each other and at the same time, should not disturb the business functionality of the system. Good theoretical knowledge, considerable analysis from various angles and great attention to the details are the essential qualities of a security architect who designs critical business applications such as 'Maximus Platform'.

References

[1] Eurocenter DDC Pvt Ltd, 2008, Eurocenter Company web site. [Online]. Available At: http://www.eurocenterddc.com [Accessed 12th July 2008]

[2] Eurocenter DDC Pvt Ltd, 2008, Eurocenter Frameworks web site. [Online]. Available At: http://ecddsee.powweb.com [Accessed 12th July 2008]

[3] ASP Definition, 2008, Wikipedia. [Online]. Available At: http://en.wikipedia.org/wiki/Application_service_provider [Accessed 12th July 2008]

[4] DMZ Definition, 2008, Wikipedia. [Online]. Available At: http://en.wikipedia.org/wiki/Demilitarized_zone_(computing) [Accessed 12th July 2008]

[5] Firewall Definition, 2008, Wikipedia. [Online]. Available At: http://en.wikipedia.org/wiki/Firewall [Accessed 12th July 2008]

[6] IDS Definition, 2008, Wikipedia. [Online]. Available At: http://en.wikipedia.org/wiki/Intrusion_detection_system [Accessed 12th July 2008]

[7] Dodds, D. et al., Relational versus multidimensional databases as a foundation for online analytical processing, [Online]. Available At: http://citeseer.ist.psu.edu/rd/0%2C638128%2C1%2C0.25%2CDownload/http://citeseer.ist.psu.edu/cache/papers/cs/30424/http:zSzzSziris22.it.jyu.fizSziris22zSzpubzSzGould_IRIS22_MDDB3.pdf/relational-versus-multidimensional-databases.pdf [Accessed 15th July 2008]

[8] .NET Application Domains, 2008, Bean Software. [Online]. Available At: http://www.beansoftware.com/NET-Tutorials/Application-Domain.aspx [Accessed 12th July 2008]

[9] ISO/IEC 27001:2005, ISO, [Online]. Available At: http://www.iso.org/iso/catalogue_detail?csnumber=42103 [Accessed 12th July 2008]

[10] Internet Information Services, Microsoft Corp., [Online]. Available At: http://www.microsoft.com/WindowsServer2003/IIS/Default.mspx [Accessed 13th July 2008]