Notes from meeting at SEB regarding Web Service and security

First, a couple of comments noted:

- Web Services are discussed as something obvious, but still (in practice) they almost do not exist today (at least not in the way envisioned)!
- Microsoft & IBM: (A note from the speech held at Stora Brännbo) Claim that they are no longer interested in Web Services outside the corporate network, only internally! This gives a hint of the problems involved in providing secure, publicly available Web Services.
- Microsoft Certificate Server – used by Banverket (?)

Being a bank, SEB naturally has doubts about introducing publicly accessible Web Services in their network environment because the organisation is an attractive target for criminals and hackers.

Christer Palm from SEB Network presented their overall firewall architecture which builds upon a layered access approach. The architecture is depicted in Figure 1.

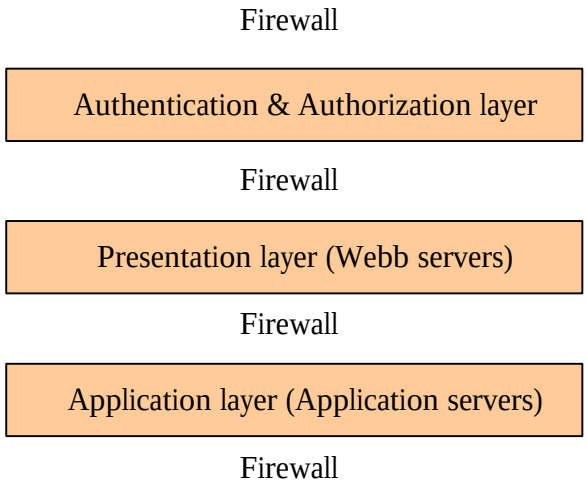

Figure 1. Firewall architecture.

As can be seen from Figure 1, the overall firewall architecture is built up from basically three layers with firewalls separating each layer. The idea is that an incoming request should be resolved as early as possible, in the outer perimeter layer. The request needs to be authenticated and authorized before it is allowed further access to the inner layers. By using the layered approach, an attacker that manages to bypass the security in place at one level should be stopped before getting access to the next level. Unauthorized access to the innermost application servers where transactions are carried out should be extremely difficult to achieve. This approach also makes it very difficult to publish Web Services that reach into the application layer.

Developers of Web Services can not use security functions built in to the service itself, because requests to these services will be filtered away before they even reach the presentation layer, unless it is properly authenticated and authorized at the perimeter layer. If a request is to be allowed to travel further across the layers there will be more work needed to be carried out

somewhere else in the architecture. The developers want to use the new technology, but there are many more people that need to have an influence and opinion regarding the solution to choose. The staffs at SEB Network are especially important in this discussion since one of their primary tasks is to control the traffic flow through the firewall.

It is considered difficult to introduce changes in the infrastructure since the infrastructure is very complicated. New products are often difficult to introduce in the existing firewall environment since they are often incompatible in one way or another. The safer a system is, the less flexible it becomes. This is a problematic trade-off that has to be dealt with.

Another problem mentioned with relying on the Internet for transport is that no one can guarantee its reliability and that it is always available 24*7. Security may be possible to achieve, e.g. by using VPN lines, but not the availability. It is possible to put demands on the ISP but beyond that it is impossible to know that your link is always up and running. Some highly critical systems are therefore built with a leased line as the basis for providing the connectivity. In these cases, the reliability is prioritized and the high cost involved is secondary.

The availability requirement (24*7) for the application servers handling transactions also makes these servers extremely sensitive to DoS-attacks, which is yet another reason why access to these servers must be extremely restrictive. A malicious attacker that gets access to a service deployed on one of these servers could easily send a huge number of requests which would most likely bring the server down, effectively resulting in a DoS-attack. IDS-systems could help to identify such an attack, but somewhere a system needs to handle the high traffic load and drop the requests which would consume valuable resources.

Security products in use

A proof of concept project has recently been carried out at SEB in order to investigate the possibility of using Web Services internally. Java-based tools like TME Glue, Axis and Web Sphere Application Developer (WSAD) 5.1 were used. SiteMinder was used in this project to provide security functions. For the internal services, basic authentication over SSL was considered to provide an adequate level of security. The server was authenticated against SiteMinder using client certificates. Thus, SiteMinder was used as a gatekeeper. It was found that Axis and Glue both include functionality for providing security, but the problem was that security issues should preferably be solved earlier, before it is possible to use security built into the service (see the discussion above).

SiteMinder (from Netegrity[1]) is the product used by SEB today to secure their webb applications. A brief description of how SiteMinder works is provided in Figure 2. SiteMinder replaces an earlier solution, built in-house. As can be seen in Figure 2, simply put, SiteMinder works by having an agent placed in a web server in the authentication and authorization layer. When an HTTP-request is received, the agent checks who initiated the request and whether that subject is allowed to access the requested resource. If the request is properly authorized, the agent includes the subject's credentials in the HTTP-header and passes the request on further down in the hierarchy. SiteMinder can perform checks at multiple levels, e.g. at the outer perimeter layer it checks out the target URL and on the lower application level it checks out the invocation of a certain method. SiteMinder can handle several ways of authentication, e.g. certificates and digipass. However, in the case of digipass, SiteMinder can only handle it if the digipass is implemented in the default way, which is not the case for SEB.

If an HTTP POST is received, a check is performed to see if the resource that is being targeted in the POST is protected. The reverse proxy simply drops POST's that are too long.

[1] www.netegrity.com

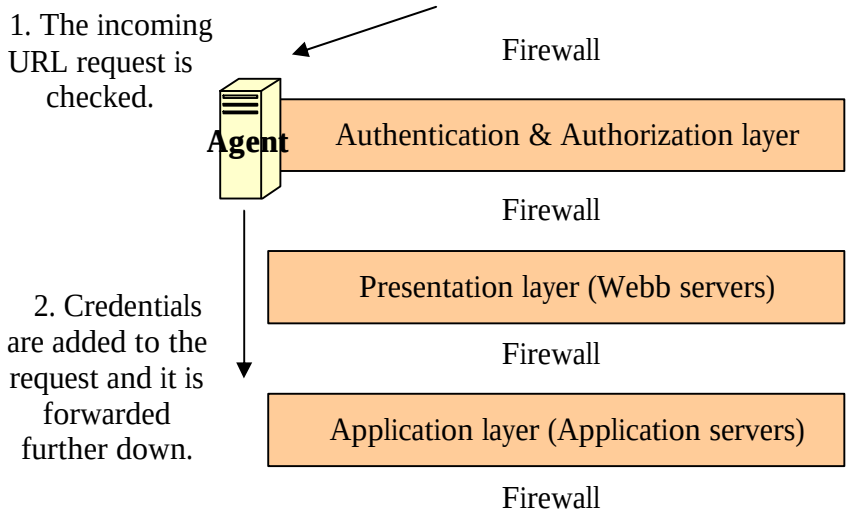

Figure 2. Description of how the SiteMinder security product for securing webb applications work.

The SiteMinder product is designed for protecting general web applications and not Web Services in particular. However, Netegrity also provides a product called TransactionMinder which is designed for protecting Web Services. SEB does not use this product at the moment but probably this is a product that they will study in more detail in the future. A coarse description of how TransactionMinder works can be found at Netegrity's website[2].

Parsing requests

Parsing the request for a Web Service with respect to what parameters are allowed was discussed. A character that is input to the service may be harmless because of its meaning at one level, but can mean a completely different (not so harmless) thing at another level, e.g. OS-level. The problem with parsing input is that it is not obvious how this can be done. How should it be done and where? Upcoming products, like XML-firewalls, have not been investigated in detail and solving the problem by installing an XML-firewall within the existing architecture was met by some scepticism. XML-firewalls provide the possibility to parse an incoming request for a service and compare actual parameters to expected input according to an XML-schema. However, not even the schemas can be completely trusted according to the SEB Network representative. The schema has been produced by a human being who can have made mistakes. The question is then if the schema can be considered trustworthy?

The EMS integration platform

The SEB CIO described their platform for front-end to back-end communication called EMS. This platform will be the preferred way to integrate with legacy systems in the future. Essentially, it is built up out of three layers; the client adaptation layer, the business logic layer and the back-end integration layer that is responsible for communicating with legacy systems and possibly also

[2] http://www.netegrity.com/products/products.cfm?page=TMhowitworks

external providers like OM. XML over HTTP is used for communication in the back-end but not in the front-end. The client adaptation layer is responsible for packaging a service so that it is available from the client application.

A problem that was described during development is that already existing services are searched for too late, when programming has already begun. One goal with the EMS platform is to increase external reuse of integration solutions. However, when discussions are held with business partners regarding what technique to use (e.g. MQ or Web Services) they often tend to stick with the old, proven solution. It is considered safer, access to personal that can administrate it is easier, it provides safe delivery etc. So the question that often comes up is; why should we choose Web Services besides maybe for checking out the new technique?

Soft certificates

Finally, a short discussion was held on the topic of soft certificates because this technique has been given a lot of critique lately. It was emphasized that security controls should be put in relation to what they are meant to be used for. Thus, in some applications, soft certificates are well suited and in others they are not. When stronger security is required, other solutions are chosen, but sometimes soft certificates provide clear benefits. The goal within SEB is to always design solutions so that they are possible to upgrade or reconfigure when new, better techniques show up.