

14 Building a Security Solution – The Wrapup

14.1	PUTTING THE PIECES TOGETHER -- BUILDING A SECURITY SOLUTION	14-1
14.1.1	<i>Overview</i>	14-1
14.1.2	<i>Establishing a Solution Architecture</i>	14-3
14.1.3	<i>Intrusion Threats Due to Networks</i>	14-5
14.1.4	<i>Intrusion Threats Due to Middleware</i>	14-6
14.1.5	<i>Intrusion Threats for Applications</i>	14-7
14.1.6	<i>Evaluating Security Versus Availability</i>	14-8
14.2	CHEATING THE CHEATERS -- HONEYPOTS AND OTHER DIVERSIONS	14-9
14.2.1	<i>What are Honeypots?</i>	14-9
14.2.2	<i>Types of Honeypots</i>	14-10
14.2.3	<i>Production Honeypots</i>	14-11
14.2.4	<i>Experimental and Research Honeypots</i>	14-11
14.2.5	<i>Commercially Available Honeypots and Issues with Honeypots</i>	14-11
14.3	WRAPUP – SECURITY SOLUTION FOR NRW	14-12
14.4	CHAP 14 -- WRAPUP – SECURITY SOLUTION FOR NRW	14-12
14.5	SUMMARY AND CONCLUSIONS	14-17

14.1 Putting The Pieces Together – Building a Security Solution

14.1.1 Overview

A good security solution approach for IS must synthesize various security threats and vulnerabilities at different layers (networks, middleware, application) of the system components and devise a solution approach. We introduced a methodology, based on SAM, in chapter 3 that went through various steps to identify vulnerabilities through attack trees and then to devise circumventions (see Figure 14-1). The first two Parts of this book concentrated on understanding the vulnerabilities within the context of organizations and the cryptographic techniques (steps 1 through 3). The focus of next two Parts was on developing circumventions and solutions based on the available technologies in the marketplace (steps 4 and 5). This part of the book is putting the pieces together by adding the issues of audit and control as a basis for continued secure operations.

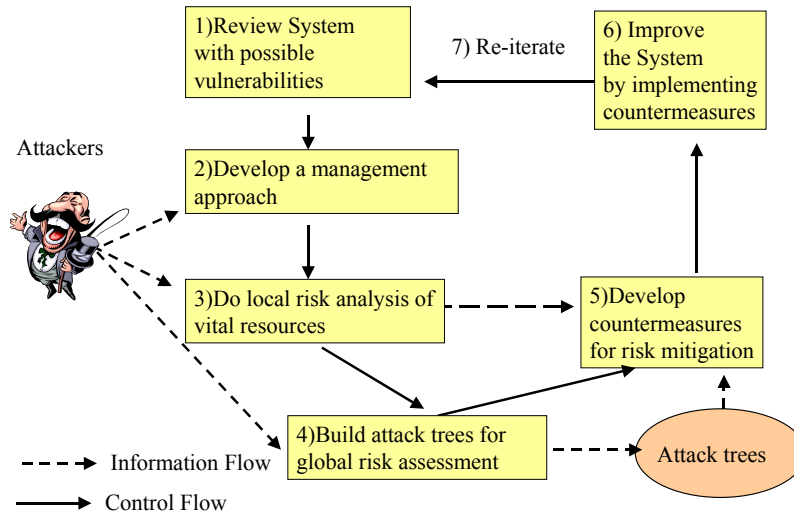


Figure 14-1: Building a Solution

Figure 14-2 shows the key elements of a security solution approach that consists of network, middleware, applications, and organizational solutions driven by a set of core security policies and architectural principles.

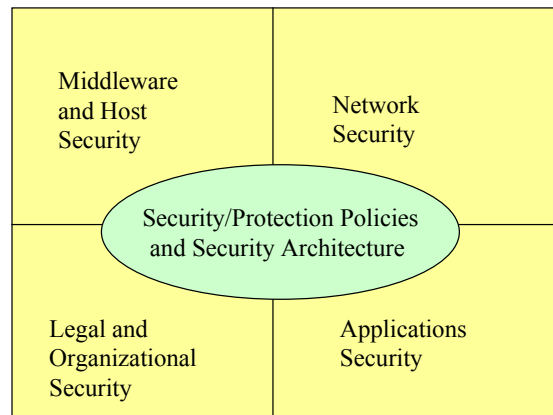


Figure 14-2: Elements of a Security Solution

The security policies need to be defined and enforced consistently across all elements of applications, middleware services, and networks. Security/protection policies based on management and organizational security approach were discussed in the previous chapter (see the sections on security requirements and management issues). Results of the security management and security policy reviews is usually represented in several documents with tables and charts. Table 14-1 shows an example of the main resources (databases, computers, networks) to be protected, the level of protection needed (based on requirements), the person who will be responsible for the protection of the resource, and the training/awareness needed. Other columns can be added to this table.

Table 14-1: Sample Security Management Decision Table

	Level of protection Needed (0 to 10, 0 means no protection needed)	Person Responsibility	Training/awareness Needed (0 to 10, 0 means no training needed)
System network	10	Joe Dumas	6
Corporate database	10	Nina Kosner	5
Product catalog	7	Sue Dasher	3

After establishment of policies, it is important to architect the enterprise system in a manner that facilitates ease of control and security. Given an overall architecture, the individual components of the system (networks, middleware, host services, and applications) need to be secured by using the technologies that we discussed in the earlier chapters. The following few sections first present an architecture approach and then summarize the analysis of these technologies in terms of networks, middleware, and applications.

14.1.2 Establishing a Solution Architecture

It is important to establish a solution architecture that facilitates ease of control and security. To develop a solution approach in modern Internet-based environments, you need to protect corporate resources (databases, programs) from intruders coming over the Internet. Another issue is: how to protect the information transmitted over the network, i.e., how to protect the information from intruders once it leaves your site and is traveling over the wired and wireless networks. The following discussion summarizes the main steps.

For corporate resource protection, the following steps are useful:

- Enforce better authentication and authorization by using the venerable IDs and passwords. In addition, public key cryptography can be used for authentication and authorization by using digital signatures, digital envelopes, and other PKI technologies.
- Use firewalls of different types (e.g., packet filtering, application proxies) at different levels (i.e., a first packet filtering firewall and then a second proxy firewall to define a DMZ).
- Make the systems themselves stronger so that even if some un-authorized user breaks in through the firewalls, he/she cannot do any damage. This involves detection of malicious code and building intrusion tolerant systems that can survive intrusions. Most of DARPA funded research is in this general area
- Architect a solution approach where the weaker systems and high risk resources are placed behind the firewalls and heavily encrypted, authenticated, etc and the stronger/low risk systems are placed in the DMZ. In addition, compensate for weakness at one level at another level. For example, systems open to wireless access are heavily encrypted, authorized, authenticated, etc.

For information transmission protection (i.e., network security), you could:

- Use AAA servers such as RADIUS for network access.

- Encrypt the messages being sent over the network. This involves VPNs, using IPsec where available.
- Use network transport level security such as SSL.
- Use application level security such as email security through PGP or S/MIME.
- Deal with wireless security for wireless LANs, cellular networks, wireless local loop security, microwave/satellite protection, etc.

As discussed previously, many open problems in security exist. For example, wireless security (e.g., WEP, GSM, etc.) is somewhat weak, AAA security is not very strong (many flaws in RADIUS standard) at the time of this writing and IPSEC is still evolving. The main task of a security architect is to know the weaknesses and design a solution that minimizes losses. For example, the following table shows some solutions for corporate assets in terms of their weaknesses and risks.

Table 14-2: Sample Security Solutions for Corporate Resources

Corporate Resources	Low risk	High risk
Weak in security	Put inside a firewall	Put inside multiple firewalls Work on improving security of system
Strong in security	Put outside the firewall or use minimal filtering	Put inside multiple firewalls

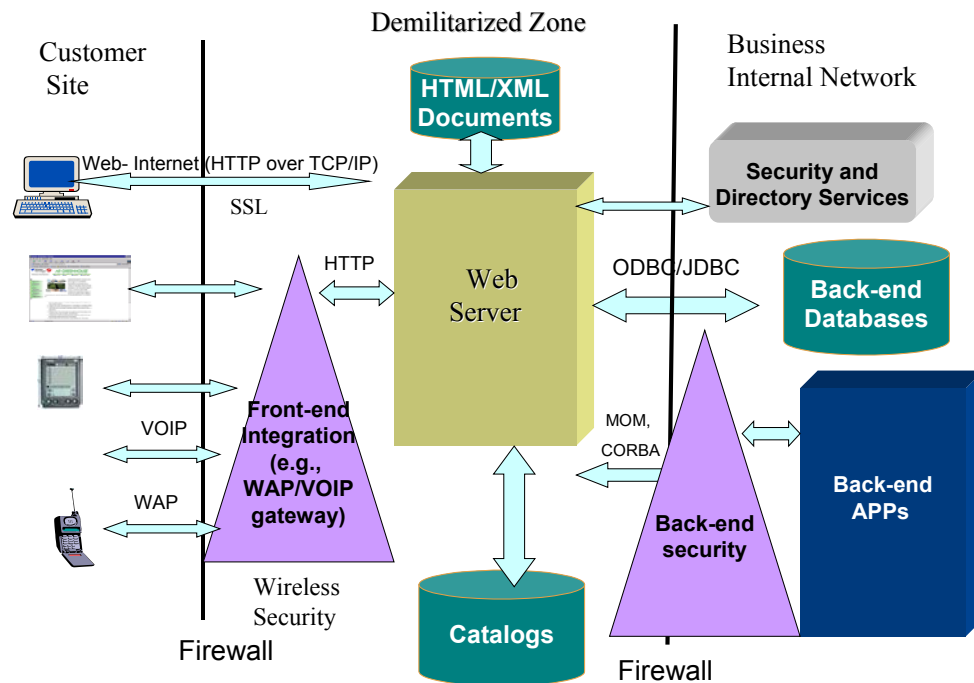


Figure 14-3: Sample Security Architecture

In summary, given the weaknesses and policies, several design principles can be used. For example, the system should be structured to take advantage of network filters ("firewalls"). It is also important that the business logic of an application runs on a server and not on an external client to minimize security compromises. The Web application

server can be used to integrate access to resources (databases, etc.), which provides greater security of the resources. A good design protects the Web server (providing presentation services) behind an outer firewall, and the remaining servers (supporting business logic) behind a second, inner firewall. This structure, shown in Figure 14-3, provides the demilitarized zone, or DMZ. In most cases, a Web server sits alone in the DMZ, handling requests from the Web and passing them along to the secure intranet network. The applications and internal business systems behind the inner firewall contain all the remaining business logic and data of the application. In addition, you can gain performance benefits by caching frequently requested data inside the DMZ rather than retrieving it from back-end systems each time it is requested. However, machines in the DMZ are known to be at higher risk. In addition to DMZ, you also need to consider security of clients. For example, mobile devices typically need another level of security before they can enter the DMZ.

After an overall architecture, the individual components of the system (databases, applications, networks, middleware) need to be secured by using the technologies discussed in this book. The following few sections summarize the analysis of these technologies in terms of networks, middleware, and applications. To highlight vulnerabilities, the discussion is centered around intrusions and a subjective ranking of various solution approaches in terms of maturity versus use. This gives us a qualitative, albeit subjective, measure of what are the major areas of concern at the time of this writing. For example, the approaches that are not mature but are used heavily are higher areas of concern. Finally, as mentioned previously, there are tradeoffs between layers of security. The application layer security is unique to certain applications (e.g., S/MIME will only protect email) but with network level security, everyone gets the same treatment (it may be desirable or undesirable).

14.1.3 Intrusion Threats Due to Networks

Table 14-3 summarizes the key results of network intrusion analysis. It shows the network services analyzed, identifies the key components with high vulnerabilities, and gives subjective rankings of immaturity and usage (on the scale of 0 to 10) of each platform. Severity of threat based on immaturity and use is estimated. The table also suggests what needs to be done to address the threats. The purpose of this table is to highlight the main areas of possible threat (the readers may or may not agree with the actual rankings in the table). As can be seen from **Table 14-3**, many network services are vulnerable and pose severe intrusion/security threats. The key result of this table is that network services, in particular those related to the wireless networks, are operating with very little intrusion tolerance in mind. Many wireless networks are vulnerable and pose severe intrusion threats.

Table 14-3: Summary of Analysis for Network Intrusions

	Components with High Vulnerability	Level of immaturity (10 means highly immature)	Level of use (0-10, 10 means used highly)	Weighted Potential Threat = immaturity x use	What needs to be done
Network	RADIUS	4	5	20	Many accounting and

Access					security problems with RADIUS. Develop workarounds.
VPN and IPSec	VPN and IPSec COTS products	2	8	16	Most IPSec tunnel mode products in the marketplace are proprietary with security flaws. Develop workarounds.
SSL/TLS/S-HTTP	SSL	1	9	9	SSL security on clients and servers need to match. Make sure that the correct level of security is used
Firewalls	COTS Firewalls	2	9	18	COTS Firewalls have many security features (need careful planning and design)
Wireless Network	WEP, Access Points	7	7	49	WEP is very weak. You must consider additional ways of intrusion tolerance for Access Points

14.1.4 Intrusion Threats Due to Middleware

Table 14-4 summarizes the intrusion and security threats due to middleware. It shows the middleware platforms analyzed, identifies the key components with high vulnerabilities, and gives subjective rankings of immaturity and usage (on the scale of 0 to 10) of each platform. Severity of threat based on immaturity and use is estimated. The table also suggests what needs to be done to address the threats. The purpose of this table is to highlight the main areas of possible threat (the readers may or may not agree with the actual rankings in the table).

As can be seen from **Table 14-4**, many middleware platforms are vulnerable and pose severe intrusion threats. The key finding is that many new COTS middleware technologies are operating with very little intrusion tolerance in mind. Many middleware platform components are vulnerable and pose severe intrusion threats. The major areas of concern identified in this report are:

- XML-based applications need special attention. Basically, XML support needs to be highly intrusion tolerant due to its explosive use. For example, intrusion of XML DTDs or Schema can have a very serious impact such as the following: 1) Trading on the hubs that use XML can completely stop if the XML DTDs/schemas are modified to invalidate the transactions. 2) Intruders can send their own “valid” XML messages that are really not valid. This could, for example, fire a missile when it should not. 3) Intruders can change XML application behavior to something very different and malicious.
- Intrusion of CORBA ORB can have a disastrous impact on CORBA applications. It is important to secure the ORB code and other resources such as name services.
- Intrusion of message oriented middleware (MOM) can have a very high impact on numerous applications that use MOM. For example, if the MOM queues are compromised, then all applications using the MOM system can produce unpredictable results.
- Intrusion of EAI Broker can destroy many enterprise applications that rely on EAI platforms. Since the EAI Broker serves as the central hub for integration, all

publisher and subscriber communication can be completely disrupted and diverted. In addition, unauthorized players can get access to sensitive data.

- Intrusion of WAP Gateway can seriously impact the WAP users. It is important to secure the WAP Gateway through high levels of security.

Table 14-4: Summary of Analysis for Middleware Intrusions

	Components with High Vulnerability	Level of immaturity (10 means highly immature)	Level of use (0-10, 10 means used highly)	Weighted Potential Threat = immaturity x use	What needs to be done
XML	DTDs/Schemas, XML documents	7	10	70	Needs immediate attention . Must make the sensitive DTDs and documents intrusion tolerant
CORBA	Orb core, Servants	5	7	35	Should protect ORB core
MOM	MOM core, Servants	5	7	35	Should protect MOM core
Next Generation Telecom Middleware	Softswitch, VOIP Gateways	7	9	63	Must consider intrusion tolerance seriously. Candidate for FRSA.
WAP Platform	WAP Gateway	6	6	36	Should protect the WAP gateway
EAI Platform	EAI Broker	8	7	56	EAI broker must be made intrusion tolerant

14.1.5 Intrusion Threats for Applications

Table 14-5 summarizes the key results of intrusion threats at application levels. It shows the application components (client-tier, server tier, and web tier) levels analyzed, identifies the key components with high vulnerabilities, and gives subjective rankings of immaturity and usage (on the scale of 0 to 10) of each platform. Severity of threat based on immaturity and use is estimated. The table also suggests what needs to be done to address the threats. As before, the purpose of this table is to highlight the main areas of possible threat (the readers may or may not agree with the actual rankings in the table). As can be seen from **Table 14-5**, many client and web tier services are vulnerable and pose serious intrusion threats.

Table 14-5: Summary of Analysis for Applications Intrusions

	Components with High Vulnerability	Level of immaturity (10 means highly immature)	Level of use (0-10, 10 means used highly)	Weighted Potential Threat = immaturity x use	What needs to be done
Client Side	XML clients,	4	7	28	XML-based security

Apps	XML/SOAP				needs to be planned
Mid Tier Apps	EJBs security	5	7	35	Should protect EJB containers
Back-end Apps	SET Transactions	2	7	35	Should use SET for payment

14.1.6 Evaluating Security Versus Availability

A wide range of security technologies ranging from public/private key encryptions to digital certificates and ACLs are currently available to address the authentication, protection, authorization, and accountability aspects of security from network to applications layers. To help us make decisions, let us revisit system protection in terms of the two dimensions: system security and system availability (see Figure 14-4). The basic idea is that a highly protected system is highly secure and 100% available. The security is provided at the following levels:

- Level 0: no security specified
- Level 1: Authorization and authentication of principals
- Level 2: Auditing and encryption (Privacy)
- Level 3: Non-repudiation and delegation

Availability is represented in terms of:

- Level 0: No replication (i.e., only one copy of the resource is used).
- Level 1: Replication is used to increase availability. The resource is replicated for a fail-safe operation.
- Level 2: FRS (Fragmentation, Redundancy, Scattering) is used. FRS schemes, discussed in the next subsection, fragment a resource (e.g., a catalog is broken into 4 fragments), replicate it, and scatter it around the network to achieve high availability and intrusion tolerance.

A protection policy can be chosen for each component of a system (e.g., networks, middleware). It should be noted that the security and availability levels are being suggested here as a basis for discussion and analysis. Different security and availability levels can be introduced, if needed.

Figure 14-4 shows the gap between where we are and where we need to be. While several efforts and implementations are underway for higher levels of security, higher levels of availability require more extensions and deployments of FRS. Extension of FRS will require extensive work and new areas of investigation. The key question is how to improve FRS and then how to imbed it in existing platforms. This is an area of future research.

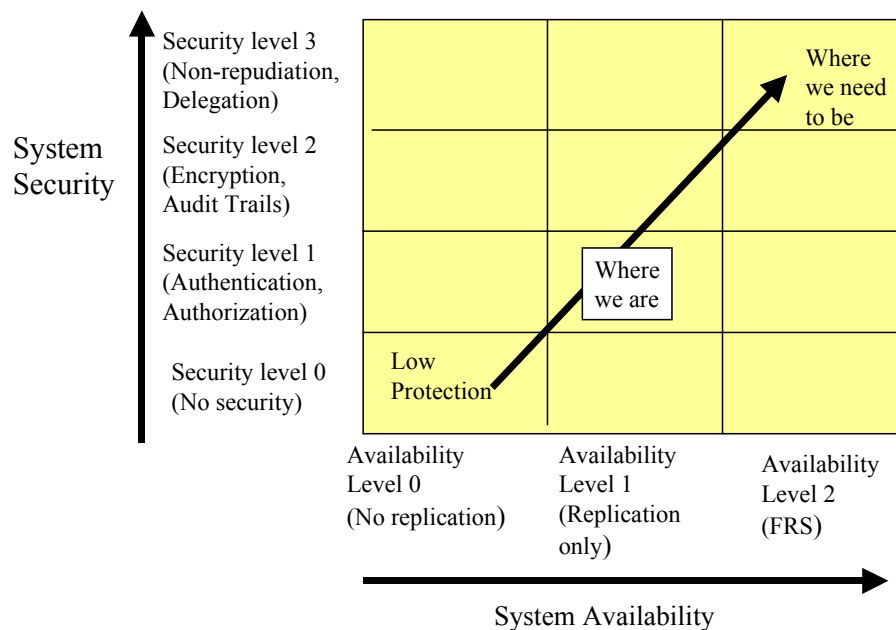


Figure 14-4: Where We Are and Where We Need To Be

14.2 Cheating the Cheaters – Honeypots and Other Diversions

14.2.1 What are Honeypots?

A question that plagues many security managers and corporate executives: is how to prevent and detect the attacks so that appropriate responses can be developed in time. A common approach is to develop honeypots that are developed especially to be probed, attacked or compromised by the intruders. These devices are called honeypots because they are expected to attract the bees, flies, and worms -- the hackers and intruders. The idea is to have attackers spend time and resource attacking honeypots, as opposed to attacking corporate systems. The attacker is deceived into attacking the honeypot, thus protecting the production resources from attack (Figure 14-5). Here are some examples:

- Fake web sites are setup by companies with misleading or wrong information to attract the attackers. These sites may pose to provide restricted, expensive or illegal information free of charge.
- Movie producers, tired of free distribution of latest movies over the Internet by "movie sites" are supporting fake movie sites. The idea is that a free-loader finds this free movie site and downloads what appears to be a good movie. But instead gets a blank tape or some junk. This is especially irritating to someone who waited for hours to download and could potentially discourage him/her.
- Music producers, also tired of free distribution of music, are increasingly resorting to fake music sites. Some of these sites also attempt to entrap the freeloaders.

Although the main idea of honeypots is to cheat the cheaters, they have become valuable resources to study the identity and behavior of intruders. Due to their role, commercially available honeypots have been developed. These honeypots allow companies to fake email sites, databases, web sites, and other resources. This section gives an overview of honeypots with their advantages and disadvantages. Most of the information about honeypots in this section is based on the articles and web site (<http://www.tracking-hackers.com>) by Lance Spitzner.

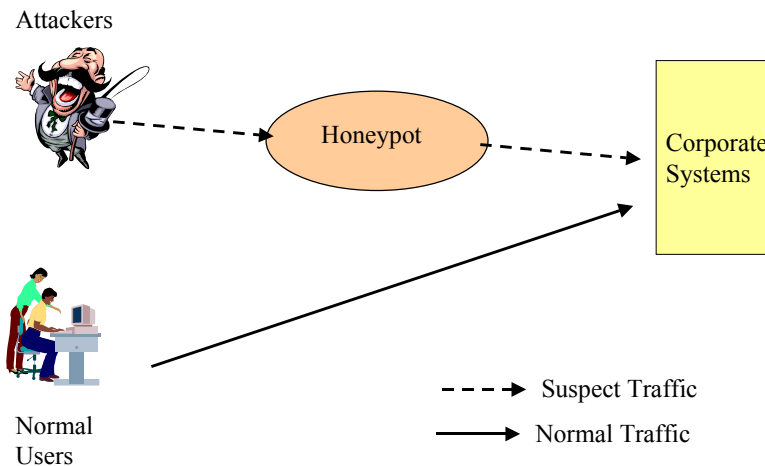


Figure 14-5: Conceptual View of a Honeypot

14.2.2 Types of Honeypots

To be valuable, honeypots must have the content and the appearance to attract large number of hackers, intruders, free-loaders and other "undesirables". Honeypots are usually designed for the following two reasons:

- **Production Honeypots.** These honeypots attempt to mitigate risk and help protect corporate resources networks. They try to confuse, irritate, discourage, divert, detect, and deal with intruders.
- **Experimental (Research) Honeypots.** These honeypots are designed to investigate and research the threats organizations face and try to understand the behaviors of the intruders. This understanding is used to develop threat models and then use them to protect against those threats.

Honeypots do not provide production services. Thus, little or no production traffic goes to or from a honeypot. Connections to a honeypot are most likely probes or attacks, and connections initiated from a honeypot most likely indicate that the honeypot was compromised. Thus, all honeypot traffic is suspect by nature.

Honeypots have some inherent advantages and disadvantages as security tools. The main advantage is that the honeypots collect focussed data that is normally of high value. Instead of mining large amounts of data that has been collected in system logs, honeypots can provide the exact information about who are the possible intruders and what do they do once they get on the system. Due to the relatively small amount of data collected, honeypots are not usually overwhelmed by the activity on the system and tend to be more stable. The main disadvantage of honeypots is that they are worthless if no

one attacks them. In addition, some honeypots may make it easier for the intruders to launch attacks and in fact may serve, inadvertently, to train some potential attackers.

14.2.3 Production Honeypots

A production honeypot, as discussed above, is used within an organization's environment to help mitigate risks. These honeypots can possibly help in the prevention, detection, and reaction of security attacks.

Production honeypots may not directly prevent the attacks. Although deception may deter and discourage some attackers, organizations must spend a great deal of their time and resources on securing their systems. Honeypots also do not prevent malicious code and viruses introduced through emails. These harmful tools may attack a honeypot, but will also attack other systems in your organization.

While honeypots add little value to prevention, they are very valuable for detection. As compared to Intrusion Detection Systems (IDSs) that attempt to guess intruders based on patterns, the production honeypots can capture actual intruders. Honeypots can complement IDSs by simplifying the detection process. Since honeypots are not intended for production activity, all connections to and from the honeypot are suspect by nature. This helps reduce both false positives (suspected intruder is not an intruder) and false negatives (a "normal" intruder is in fact an intruder) that typically plague the IDSs. But honeypots cannot detect the intruders that do not visit them. Thus, honeypots and IDSs complement detection systems -- IDSs can detect some intruders while the honeypots catch the others.

Production honeypots can also help enterprises to recover from the attacks. For example, consider an organization that has three web sites and one honeypot. Let us assume that all four are compromised by an attacker. Naturally, the company wants to bring the three production sites online quickly. However, the hotspot can be used to analyze what failed, what damage was done, who was the attacker, etc. These lessons could then be applied to the remaining web servers, allowing us to better identify and recover from the attack.

14.2.4 Experimental and Research Honeypots

Research honeypots are designed to answer questions such as what are the threat, why do they attack, how do they attack, what are their tools, and possibly when will they attack? Good research honeypots allow the researchers to watch the attackers in action, to record step-by-step as they attack and compromise a system, and what they do after they compromise a system, (e.g., communicate with other intruders or upload a new tool kit). Research honeypots can also show how to capture and contain automated attacks that spread from one site to multiple sites. The lessons learned from research honeypots can be used to build better IDS tools.

14.2.5 Commercially Available Honeypots and Issues with Honeypots

Different types of honeypots are commercially available at present. Some of these honeypots are simple software packages that emulate basic services, such as http, ftp,

telnet, and mail. Thus any intruder who tries to use these services is misdirected or trapped. BackOfficer Friendly and Specter are examples. On the other hand, many honeypots such as Honeyd are sophisticated systems that can emulate hundreds of different operating systems and computer systems. In between, there are many homegrown and commercial honeypots that are designed for special purposes.

Honeypots while very useful, raise some legal and privacy issues. Naturally, entrapment is a possible issue because an attacker can claim that he/she was fraudulently induced to use the honeypot. This claim could especially be made if the intruder was to be prosecuted. Privacy of the files and messages on the honeypot left by a potential intruder is another issue. In particular, what if an innocent person walks into a honeypot and leaves her files on the honeypot. Who owns those files? The statutes concerning communications privacy -- the Electronic Communication Privacy Act (18 USC 2701-11) and federal Wiretap Statute (Title III, 18 USC 2510-22) -- need to be considered. Similar privacy laws in other countries exist and must be considered if you are implementing honeypots in international settings.

To summarize, honeypots are effective tools in the study, analysis, detection, and possibly prevention of attacks. Honeypots, however, do not solve an organization's security problems. Organizations need to pay a great deal of attention to securing their networks, application, and database. Honeypots can be used as an extra tool, but by no means as a replacement of good security measures. The website (<http://www.tracking-hackers.com>) has a great deal of information on this topic.

14.3 Wrapup – Security Solution for NRW

14.4 Chap 14 – Wrapup – Security Solution for NRW

Let us wrapup by using the NRW case study that we started in the beginning of this book. We will illustrate how the various components are combined to provide a secure architectural solution for NRW. Figure 14-6 revisits the security stack that provides the technical solution for NRW. This stack shows in more detail the key security players (physical network security, IPSec, SSL, Kerberos, S-HTTP, S/MIME, PGP, CORBA security, SET, etc.) and the interdependencies between the various components of a system.

NRW has gone through a major architectural effort so that the application architecture uses a logical n-tier Web application model with a thin HTML-based client that uses Java, XML, and Enterprise Java Beans (EJBs) for access to the existing customer account database. A corporate 802.11 wireless Intranet has been installed and users can also access customer account data from cellular phones with GSM. The company wants to allow its customers to access and update their account information and use some of the firm's Internet tools to experiment online. This customer account/problem information can also be accessed/manipulated by authorized employees of the corporation.

The corporation has identified several main security risks and defined mechanisms to address them. Here is a summary of a few:

Risk1:

- Risk statement - Information flow, including passwords and account data, over the Internet is not secure and may be stolen.
- Mechanisms to address the risk- Ensure there are secure communications between the end user and NRW. Make sure that all network traffic between NRW and their customers is protected using SSL at a minimum. For dialup users, support PPP over a VPN.

Risk2:

- Risk statement- Hackers may attempt to access the NRW system by trying user ID and password combinations to impersonate an existing customer.
- Mechanisms to address the risk - Ensure that the system can determine and verify user identity properly. Implement strong mutual authentication using PKI. Provide token cards/smart cards with certificates to all customers who sign up for the service and encourage their use. Provide X.509 certificates on a browser as an alternative.

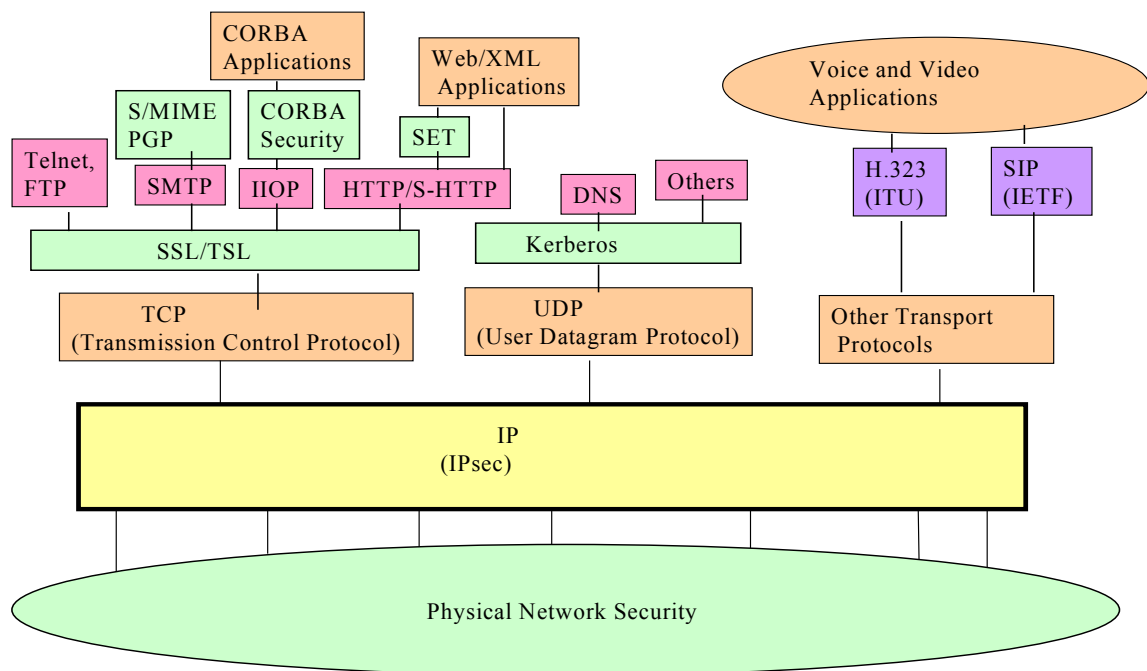


Figure 14-6: Security Stack – Revisited

Risk3:

- Risk statement- Hackers may try to attack and penetrate the NRW network, and infect the system with a computer virus or malicious active content, etc.
- Mechanisms to address the risk - Protect the enterprise network and, where possible, the customer end user system from intrusion and attack. Provide antivirus software for end users who sign up for the service. Install antivirus and intrusion detection software in the enterprise. Implement a DMZ between the company intranet and the public Internet.

Risk 4

- Risk statement -- Wireless access from the corporate Intranet as well as remote users needs to be fool proof.
- Mechanisms to address the risk -- Wireless security is weak at lower levels (e.g., WEP problems). It is important to encrypt the wireless traffic by using IPSec and use strong higher level security (e.g., WLS) for the users who are coming through the wireless network.

Risk 5

- Risk statement -- The company is concerned that the new system is based on many emerging COTS middleware products that are weak in terms of security. The company is especially concerned about the following vulnerabilities:
 - Directories that contain addresses of participating applications in NRW may be contaminated, or modified, so that the traffic between the participants stops even though the network, the operating system and computing hardware are fully operational.
 - XML Document Translation Definitions (DTDs) used in XML-based messages exchanged are corrupted so that all XML messages become invalid thus halting the message exchanges because every transaction becomes invalid.
 - Someone contaminates the EJB (Enterprise Java Bean) container disabling the entire application.
 - Call agents for VOIP (Voice Over IP) and other "next generation network" applications are corrupted to allow unauthorized routing of calls and eavesdropping with obvious undesirable consequences.
- Mechanisms to address the risk -- A wide range of security measures for XML, EAI, EJBs, and VOIP need to be instituted by using encryption and authenticating the users who can modify the XML directories, EAI hubs, EJB containers and VOIP gateways.

As stated previously, the application architecture uses a logical n-tier Web application model with a thin HTML-based client that uses Java and Enterprise Java Beans (EJBs) for access to the existing customer account database. Based on this overall architecture, and the risk analysis, the following security-related design decisions were made:

- The system is designed to fit into the DMZ (demilitarized zone) model. The application's presentation logic is deployed within the DMZ and the application's business logic is deployed within the intranet.
- Within the corporation, protect wireless data traffic through IPsec by using one of the commercially available products. An example of such a product is the Nortel Extranet Access Switch that connects wireless users to a corporate network. The wireless users install and configure the Extranet Access Client, a Windows application, to create and store connection information for tunneling into the Extranet Access Switch. The Extranet Access Client uses the IPsec protocol with the ISAKMP/Oakley Key Exchange protocol to authenticate and secure an end-to-end connection into a remote network.
- Heavily protect the sensitive XML directories, EJB containers and other new services by using authentication and encryption. This can be done through SSL.
- It may be advisable to fragment, replicate and scatter (FRS) some sensitive corporate data so that even if a hacker breaks into the corporate databases, they cannot access all information.

- All information about users and groups is stored in a centralized directory service, deployed in the intranet, to decrease complexity and make the application easier to administer when users are added or deleted.
- A centralized authorization service is used to make it easier to define and manage the permission policy for access to programs, data, and other resources. It is deployed in the intranet. A trust relationship among the systems is used.
- The end user is required to sign on (log in) to the system once and only once. All system interaction is transparent to the end user. Credential mapping is used, where necessary, to implement single sign on.
- NRW will not issue certificates. A 3rd party Certificate Authority that implements Public Key Infrastructure, or comparable software, is used for this function.

Next step is to define and configure the principals and objects for each system, and their permissions, in a consistent fashion by using a Security Manager (SM) such as IBM's Security Policy Director. The firewall systems are configured on each side of the DMZ. The outer firewall (router) allows only HTTP/HTTPS protocol flows, and the inner firewall allows only those protocols needed to access the Internal NRW systems.

Policies and controls are needed to assure that the system keeps running as specified. In addition, security audits described in the previous chapter should be conducted on an ongoing basis to assure that the policies and controls are being properly complied with. Honeypots may also need to be deployed to divert the potential attackers.

Figure 14-7 shows how the security architecture, when combined with the application architecture, results in a trusted architectural solution. The end-to-end flow sequence for access from an external customer follows:

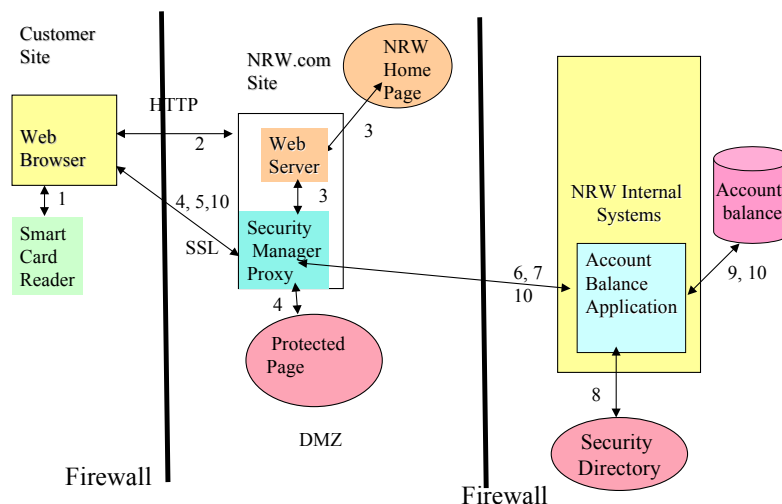


Figure 14-7: Security Flow Example

1. Sue is a NRW customer (user). She can access the system by using a variety of methods:

- a) She uses a Smart card. She inserts her Smart Card in the Smart Card Reader attached to her PC and enters her PIN number to enable her system. Sue then dials into her ISP for connection to the Internet and starts her Web browser.
 - b) She uses a VPN and a token card over a dialup network. In this case, she first dials the VPN supplier number and then gives her token card number along with a user ID to access the corporate network.
 - c) She wants to use her laptop in wireless corporate Intranet. In that case, she uses her Extranet Access Client to access the corporate network through the Extranet Access server for wireless.
2. Sue accesses the NRW home page by typing <http://www.xcorp.com>. The HTTP request flows through the NRW outer firewall/router to the Security Manager (SM) proxy.
 3. The SM proxy inside the DMZ receives the HTTP request and determines that the NRW home page is not protected, so the Web page is sent to Sue.
 4. This home page includes a link to a protected page. By linking to this page, an SSL session is established between the browser and the SM proxy. As part of SSL processing, and to identify Sue to the SM proxy, the browser accesses Sue's certificate and private key from the smart card (for example), which was activated in Step 1.
 5. The SM proxy sends Sue's certificate to the SM, to establish her logon. The SM proxy then uses its copy of the SM access control list (ACL) to determine whether Sue has the permissions needed to access the protected Web page that lists customer applications.
 6. The "Welcome" Web page is sent to Sue. It contains links to available applications. Sue clicks on the "Account" application link which sends an HTTPS request to NRW.
 7. The SM proxy ensures that Sue is authorized to obtain her account information. Once authorized, Sue's user credentials to NRW-Accounts-Application are obtained and the HTTPS request is forwarded to NRW-Accounts-Application, including those credentials.
 8. NRW-Accounts-Application issues a call to Security Directory running behind the inner firewall to authenticate Sue. This request flows through the inner firewall. This establishes Sue's logon to NRW-Accounts-Application.
 9. NRW-Accounts-Application evaluates if Sue is authorized to execute the needed method. Several additional checks are made to make sure that Sue is authorized to access this information. Finally, an SQL query is issued and run after the database manager authenticates the credentials and authorizes access.
 10. The results of the query are returned. The data is passed back through the systems to NRW-Accounts-Application running in the DMZ where the data is formatted into a Web page which is sent to Sue over the SSL session.
 11. Sue now places an order. A similar process takes place. The only difference is that the order processing application is implemented as an EJB that accepts XML input and also accesses the account application. The XML repository is checked to make sure that only authorized users can access it. In addition, the EJB containers are protected so that they cannot be modified.

Figure 14-8 shows a more detailed physical architectural view of the NRW system that displays the application, middleware, local software, as well as network components. This view is a refinement of the views presented in the NRW case studies so far and can serve as the basis for establishing controls and audits.

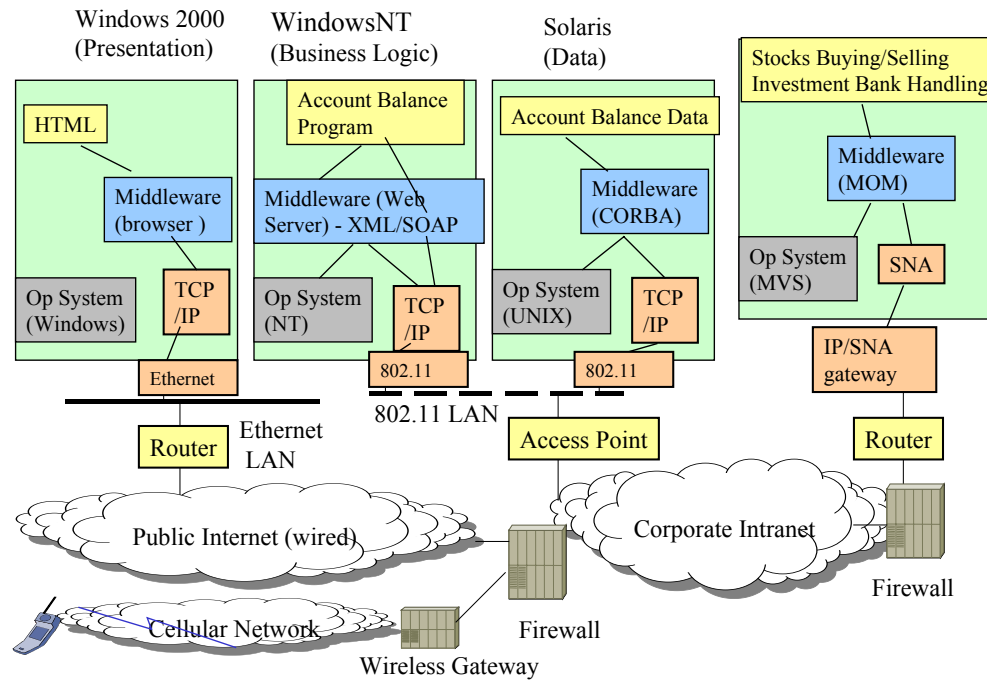


Figure 14-8: Detailed Physical Architecture of the New NRW System

14.5 Summary and Conclusions

Using Figure 14-9 as a framework for discussion, we have attempted to present an architectural view that describes how different management and technical approaches can be used to build a security solution in the modern digital environments. We have reviewed security at network, middleware, and distributed application levels. For each level, we have discussed the technologies and the areas of vulnerabilities are identified in terms of use and immaturity of security solutions. An example has attempted to show how various pieces fit together.

As we have seen in various chapters, a wide range of security technologies from public/private key encryptions to digital certificates, and ACLs are currently available to address the privacy, integrity, authentication, authorization, and accountability aspects of security from network to applications layers. Although many technical solutions are commercially becoming available, they are not flawless. Numerous cases have been published showing the weaknesses of RSA, SSL, and several other techniques. Although

the situation is improving with better implementations and also with longer key sizes, several areas of research are being pursued. Due to the importance of cryptography, a great deal of research work at present is concentrating on newer and better cryptographic techniques. From an operational point of view, security solutions must balance between security and performance/availability requirements. An interesting area of research work is extension of FRS (fragmentation, replication, scattering) to satisfy the combined security and availability requirements.

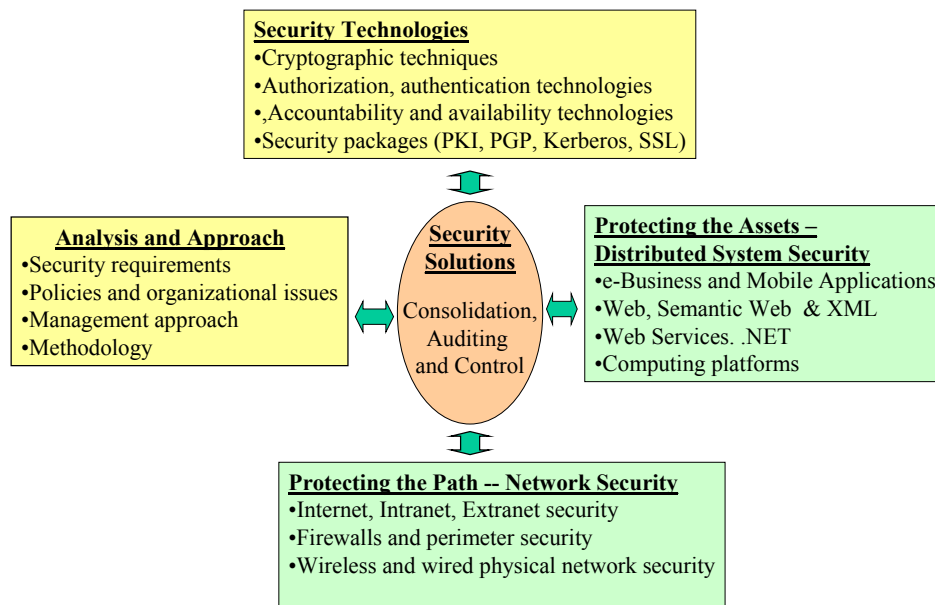


Figure 14-9: Building a Security Solution – The Framework