12 Applications Security: Protecting e-Commerce and Mobile Applications

12.1 Introduction

Security of applications and associated data is naturally of paramount importance to modern corporations. In this digital age, most of the business value comes from its applications such as web-based advertising, online purchasing, customer relationship management, and automated supply chain management systems. To be successful in the digital age, companies must streamline and integrate applications that support customer purchases, payment systems, and interactions with suppliers. These applications are key players in modern businesses because they can enable, and if not handled properly, disable business strategies and business designs.

Before proceeding, let us define applications. According to Webster, an application "puts to use especially for some practical purpose". For the purpose of this chapter, we are concentrating on computer applications that put to use computers for the practical purpose of supporting businesses. Specifically, we are primarily interested in the class of computer applications that are business aware, i.e., *we are interested in business applications*. Thus, for our purpose, we define an application as follows:

> **Definition**: An application system, commonly referred to as application in this book, is a business application that uses computers to support businesses, thus it is business aware and represents the business aware functionality and data.

Business in this context could mean any type of business such as manufacturing, aerospace, healthcare, finance, or telecom. Examples of such applications are airline reservation systems, inventory control systems, financial planning systems, material handling systems and the like. These applications are used by organizations to gain/retain competitive edge, reduce costs, and improve management decision making and reside above the networks and middleware/platform services (Figure 12-1).

This chapter starts by examining security of modern applications in terms of multi-tiered distributed applications security. This allows separation of issues and also helps in examining the interrelationships between the network, platform, and application security. The security issues unique to e-commerce, e-business, and mobile commerce are then discussed. The chapter concludes by reviewing email security and discussing various viruses and worms.

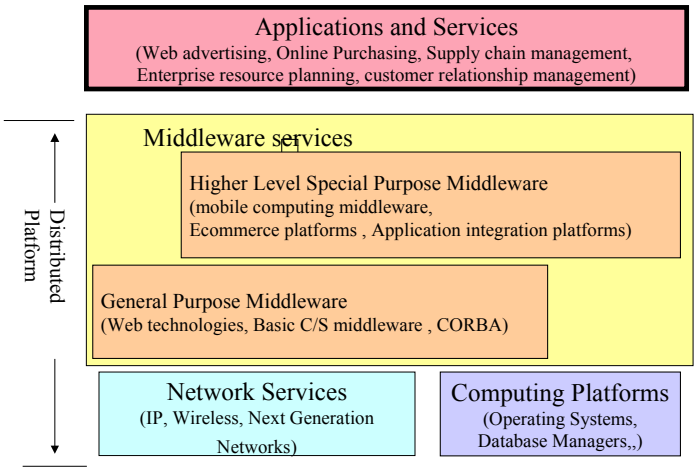


Figure 12-1: Applications and Distributed Computing Platforms

Chapter Highlights

- The business applications are a valuable asset of the modern digital enterprises and need to be protected against assaults.
- Security of modern applications can be viewed in terms of multi-tiered distributed applications security. This allows separation of issues in terms of client-side security, middle-tier (mostly Web) security, and back-end (including legacy system) security. This view also corresponds to the modern component-based architectures that are at the foundation of .NET and J2EE.
- Basic e-commerce security can be discussed in terms of online purchasing security. Secure Electronic Transaction (SET) is a key enabling technology for online purchasing because it supports secure credit card processing.
- e-Business applications can be secured by protecting:
  - Business to consumer (C2B) applications such as online buying and web publishing.
  - Business to employee (B2E) internal business applications such as enterprise resource planning (ERP).
  - Business to business (B2B) trade conducted between partners directly (e.g., supply chains) or through an intermediary (e.g., an emarket).
  - Consumer to consumer (e.g., collaborative computing) and consumer to data (e.g., e-business intelligence) applications.
- Mobile applications allow hand-held devices to access and use a wide range of databases and applications for e-banking, retail payment, brokerage, and e-business. Security of these applications can be discussed in terms of:
  - *Mobile* Enterprise Business Applications (MEBAs) that add the mobility dimension to EB applications such as ERPs, SCMs, CRMs, etc. .
  - Mobile Commerce (M-Commerce) applications that allow cellular phones and PDAs to search the Internet, access data and information, and conduct purchasing or business transactions.
  - Voice Commerce (V-Commerce) is gaining importance to support users who want to use telephones and other voice-driven devices for conducting e-commerce.

- Positional Commerce *(P-Commerce)* is becoming popular to provide support to the customers based on their geographic position (e.g., give you information about deals in the Atlanta area when you are in Atlanta).
- Mobile agents security is very challenging. These agents roam around the network looking for information and bargains on behalf of the customers. Most mobile agents at present are Java applets that go from one computing system to another.
- Email can be secured through PGP or S-MIME.
- Malicious code such as viruses and worms are serious problems at present. In many cases, malicious code is introduced through emails.

12.2 Distributed Application Protection – General Considerations

12.2.1 Overview

Distributed applications have a multi-tier architecture consisting of client tier (typically presentation logic on a browser), middle tier (typically business logic on a web server), and back-end tier (typically large enterprise databases and applications on mainframes). This multi-tier application architecture introduces several security issues that span networks and middleware services across multiple platforms. Many security services for applications are becoming available. For example, a Web application server can be used to integrate access to resources (databases, etc.), which provides greater security of the resources.

A good security 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 12-2, is known as a demilitarized zone, or DMZ. This figure shows how a browser accesses a purchasing system at a web server that in turn accesses back-end systems. The interactions go through networking (TCP/IP stack, routers), operating systems, and firewalls. 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. DMZs provide several benefits. For example, 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 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.

Using Figure 12-2 as a framework, the balance of this section briefly reviews security issues at the client, middle tier, and back-end system tiers. This framework is then used to discuss security of e-commerce, e-business, and mobile applications. This framework

also corresponds to the modern component-based architectures based on .NET and J2EE and thus can be used to analyze most modern applications.

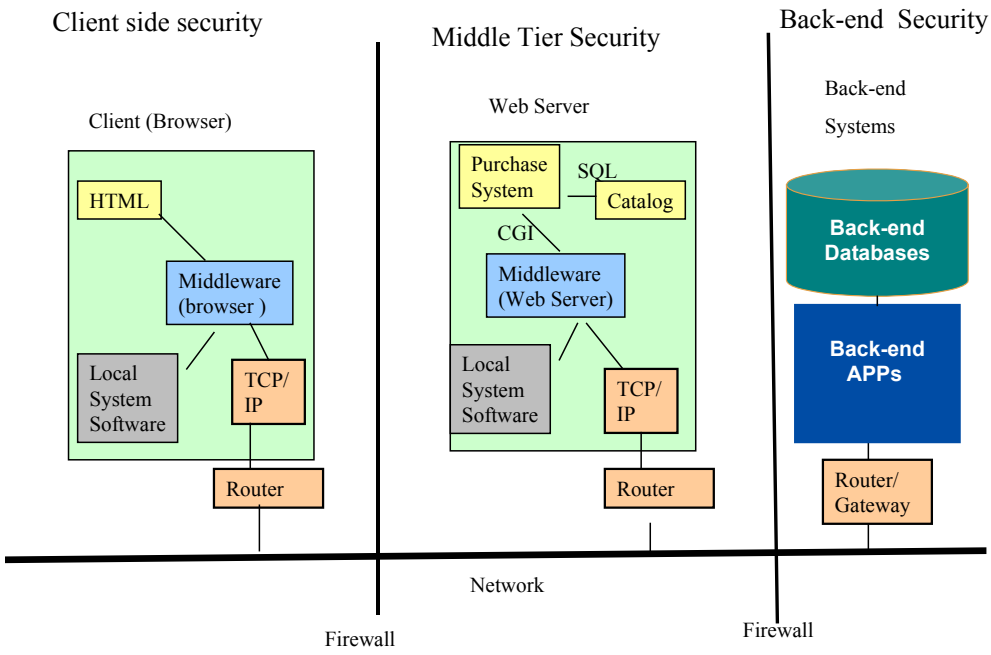


Figure 12-2: Multi-Tier Distributed Applications Security

12.2.2 Tradeoffs between Application and Lower Level Security Issues

Let us briefly review the tradeoffs between application and lower level security issues before proceeding. In particular, how do network security approaches such as VPNs/IPSec, middleware security such as SSL, application security such as PGP, and Web Services security such as SAML interrelate with each other.

Figure 12-3 shows the main approaches. If you use the network level security by employing VPN/IPSec (Figure 12-3a), then it is transparent to all applications. Since almost all applications run on top of IP, all communications between application clients and servers are protected by using this option without any changes to the applications. Another option is to move security to a higher level, i.e., above the TCP layer as shown in Figure 12-3b. The most widely used option is the use of SSL (Secure Socket Layer). SSL could be made transparent to the applications by using it as part of the TCP/IP stack or it can be packaged in specific applications. For example, SSL is currently packaged with web browsers and servers. Application specific security services such as PGP, S/MIME, and SET (Secure Electronic Transactions), shown in Figure 12-3, are used to protect specific applications.

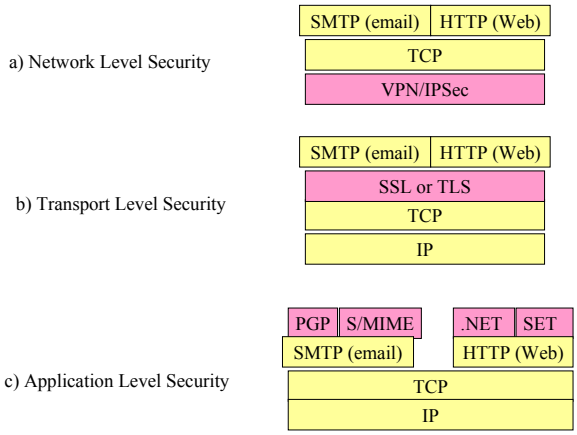


Figure 12-3; Approaches to Web Security (Based on Stallings [2000])

12.2.3 Client Side Security

Typical threats on client side security are based on active content that appears as:

- Helper applications and plug-ins that are used to handle special programs (e.g., Powerpoint) that are not shipped with web browsers
- Java script and VB script that are used to imbed executable code in HTML pages
- XML processing on the client side that uses Extensible Stylesheet Language (XSL) to convert XML to HTML
- Java applets that can do a variety of tasks on the browser (e.g., show video clips, draw graphs and charts)
- ActiveX control applets that control the display of MS Desktop Services
- CORBA clients that issue calls to remote CORBA objects
- XML/SOAP clients that access remote resources in the Web Services environment
- Browser-side cookies that keep track of the activities at browser sites
- Other client-side tools such as Macromedia tools for various tasks

Any of these programs can be compromised, replaced, and/or modified to show render undesirable behavior. Due to space limitations, it is beyond the scope of this book to discuss protection of this content in detail. We will discuss important issues as we go along.

12.2.4 Web Server Tier (Middle Tier) Security

There are many serious issues on the web server (middle tier) sites. Examples of the issues are:

- HTML content that may be accessed and modified by unauthorized users.
- XML content that may represent customer data may be accessed and modified by unauthorized users.
- CGI scripts, Servlets, JSPs (Java Server Pages) and ODBC/JDBC drivers may be modified to access different databases and applications.
- EJBs (Enterprise Java Beans) containers can be contaminated to disable EJB applications.
- Web Server and SOAP "listeners" can be modified to invoke malicious code.

- Server logs can be accessed to review confidential information and even modified if not protected properly.

When data must travel outside of a secure system environment, it needs to be protected so that the policies governing its use cannot be violated. Secure communications, ensuring data privacy, data integrity, and origin authentication are an important aspect of information protection. This, as discussed previously, has to be handled through network level protection (VPN) and/or firewalls.

A common attack on Web-based is launched by using weaknesses of the server side programs. These weak programs can be test programs that were left on the system inadvertently, a default CGI script from an application server, or a program with faulty logic that was never debugged and corrected. The attackers invoke these weak server side programs by using browsers. The attackers either try to gain privileged status through these programs and/or install trojan horses or backdoors that can be used by the attackers later to enter the system unnoticed.

The best way to defend against such attacks is not to leave untested and flawed code on the machines. In particular the server side programs need to be carefully watched because they can be invoked by users from the Web browsers anywhere on the Internet.

12.2.5 Web Services Security and SAML (Security Assertion Markup Language)

Web Services (WS) Security is an area of considerable activity at present and the interest in this activity is growing due to the popularity of Web Services. Web Services security models, as discussed in the previous chapter, integrate and unify several popular security models, mechanisms, and technologies (including both symmetric and public key technologies). The specifications build upon foundational technologies such as SOAP, WSDL, XML Digital Signatures, Kerberos, XML Encryption and SSL/TLS. Security Assertion Markup Language (SAML) is at the core of the WS Security architecture. SAML uses XML to exchange security information in the form of assertions. Since SAML does not directly provide message integrity or confidentiality; it relies on XML Signature to protect integrity and on SSL (Secure Sockets Layer) and TLS (Transport Layer Security) for confidentiality. We briefly reviewed WS security and SAML in the previous chapter. Additional information can be found from (www.oasis-open.org) and (xml. coverpages.com/saml.html).

12.2.6 Back-end System Security – Protecting the Legacy Systems

Information must be protected at the back-end sites where it exists in large databases. Access control (allowing authorized users to access needed data) protects data at various sites. Most database managers have security features that allow only authorized users to access needed data. In some cases, data is encrypted and stored for additional security. The topic of back-end system protection is well discussed under the general heading of "host security" [Oppliger 2000] and typically includes discussion of operating system and database security. For our purpose, it is important to note that back-end systems should-be typically behind DMZ walls.

Security of legacy applications is important because many back-end applications are considered to be "legacy". Let us briefly review what legacy systems are. According to Webster, legacy is "something of *value* received from an ancestor or predecessor or from the past". The term legacy is being used in several contexts such as "legacy LANS" (e.g., Ethernet LANs), "legacy operating systems" (e.g., PC DOS), and "legacy management styles" (e.g. "Taylorism" -- the Scientific Management Movement by Frederick Taylor [Taylor 1911]). Some people use the term legacy to indicate anything that is not fashionable. For example, at present the .NET enthusiasts are labeling all non .NET applications as legacies. We will use the following definition, based on the Webster dictionary:

> **Definition.** A legacy application is an application of value inherited from the past. (typically 5 years or beyond).

The two keywords in this definition are of value (critical to the business) and inherited from the past (typically 5 years or beyond). Many debates about legacy applications took place in the mid 1990s when client/server and Web technologies became popular. Many legacy applications of interest at present are IBM mainframe-based that were built in the pre-PC and pre-Web era (i.e., late 1980s to early 1990s). These systems provide back-end support and typically use IBM mainframe technologies such as IMS-COBOL-3270-CICS (see Table 12-1)

Table 12-1: Typical Legacy Mainframe Technologies

	Data Technologies	Program Code	User Interface	TP (Transaction Processing) Monitor
Early 1970s	Flat files (sequential, indexed sequential, VSAM, random)	Assembler and COBOL	Character-based	None or Homegrown
Late 1970s	Non-Relational Databases IMS (Very few use IDMS, Total, and System 2000)	COBOL and PL/1 (Assembler is rare)	3270 Screens	CICS or IMS-DC
Mid to late1980s	Relational Databases such as DB2 (Very few use Oracle on MVS)	COBOL and PL/1	3270 Screens	CICS or IMS-DC

Legacy applications contain very valuable information that is embedded in legacy databases/flat files and application code. In many cases, legacy applications are the only source of years of business rules, historical data, and other valuable information. Access to this information is of vital importance to new and emerging tools and applications. These tools and applications reside on a variety of platforms (PCs, Macs, UNIX workstations) that access legacy applications through different network technologies (TCP/IP, SNA, Novell IPX/SPX). This problem has been discussed widely in the

literature under the headings of "legacy data access", "universal data access", "corporate data access", "open data access" etc., and has been ranked among the top corporate problems at present. In general, many industries are experiencing rapid changes and new competition to which enterprises must respond quickly and effectively. In order for enterprises to deploy services rapidly, they must be able to effectively access their existing information and invoke new functionality for new services. For example, many new services still need customer information that is typically stored in legacy systems.

Legacy information access spans a wide range of issues and approaches. The needed information is stored in diverse data sources (hierarchical, network, or relational databases, in addition to the flat files such as sequential, indexed, and direct files), program code and command files. Most of the legacy information is at present stored in the hierarchical/network databases and/or flat files, with some examples of DB2 relational databases. We view legacy information access as a class of enterprise information access problem. Instead of dealing with the legacy information access problem as a unique and isolated problem, our view has the benefit that the concepts and tools to access an enterprise information source (e.g., DB2) can be used even when an organization cannot decide if DB2 is legacy or not.

Security of legacy systems raise typical issues of privacy, integrity, authorization, authentication, and accountability. However, there are some reasons for comfort:

- Legacy application and databases contain information that is typically boring to the average hacker.
- The skills to break into legacy systems are not common because most hackers know very little about old IBM mainframe-based systems.
- "Security by obscurity" works well because most of these applications lack structure and documentation is rare.

The most pragmatic approach to secure legacy systems consists of the following steps:

- Make sure that the lower level issues of firewalls, networks, and platforms have been properly handled to provide a secure environment for these systems.
- Place the legacy applications in highly protected areas such as behind DMZ firewalls.
- Do your best to use the existing security features and add others if needed. Since many legacy applications are IBM mainframe-based, security facilities such as RACF (Resource Access Control Facility).
- Carefully examine and secure the paths (the links) between the legacy applications. This could be done by using typical network security approaches such as VPN.

12.3 Basic e-Commerce Security and SET

12.3.1 Why e-Commerce Security is important

Simply stated, *e-commerce (EC)* is buying and selling over the network (mostly Internet). Due to this, online purchasing is at the core of e-commerce because it represents online buying/selling through an online catalog. At a very basic level, e-commerce security can be discussed in terms of online purchasing security. This includes privacy and security issues related to consumers, buyers, and suppliers engaging in on-

line trade. In addition, back-end systems for inventory updates and credit checking are brought into the picture.

As shown in Figure 12-4, the purchasing process consists of several steps that can be viewed in terms of pre-purchase, purchase consummation, and post-purchase activities. In the pre-purchase activities, the users browse through various sites, compare prices, and select the online-merchants they want to buy the goods from. Naturally, Web and the Internet have had the most profound impact on these activities. In the purchase consummation activities, the user may use a shopping cart and place an order by using a payment system. Naturally, the e-commerce payment systems play an important role in this area. The post purchase activities involve the classical "back-end" systems that handle payments settling, shipping and receiving, etc. Many of these applications are legacy applications that have been around since the 1970s and 1980s.

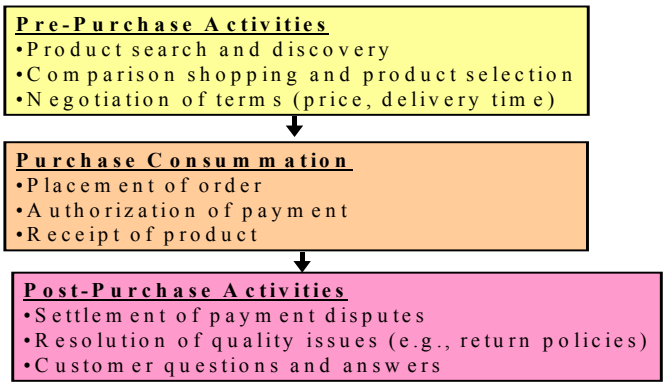


Figure 12-4: Purchasing Steps

e-Commerce applications such as online purchasing share the same type of security risks as many other distributed applications. However, there are some unique issues:

- Online purchasing involves money and very few things excite the criminals more than stealing money. By stealing credit card information, the intruders can buy things -- an appealing opportunity.
- Privacy issues arise due to buying and selling over the networks. For example, if a customer buys some drugs by using a credit card, then this information can be misused by an intruder.
- Multiple players look at some information that they do not need to see. For example, in a drug purchase, the credit card company knows what you bought. In addition, the merchant keeps a record of your credit card information. Neither party needs this information.

The difficulty in securing online purchasing systems is that they span new Web technologies as well as old legacy systems. Specifically, online purchasing involves a large number of Web based systems that allow users to search company catalogs for certain price ranges and then place orders for chosen product(s). These systems also need to support mobile users. In addition, the order processing, inventory control, payment, and shipping/receiving systems are employed. All these systems need to work together to satisfy the demands of online buyers and sellers. Due to this demand, several specialized middleware services have become available to support mobile computing and online

purchasing and are also being packaged with other infrastructure services to form ***"Middleware Platforms".*** Examples of these platforms are ***e-commerce Platforms*** such as IBM's Websphere and Microsoft's Internet Commerce platform. These platforms introduce their own security risks that were briefly discussed in the previous chapter.

Let us briefly review some common examples of online purchasing such as Web storefronts and virtual shops. This will get us started in e-commerce security. Additional e-commerce security issues involving emarkets will be discussed under e-business security in the next section.

Web Storefronts. Web storefronts use the Internet to market and sell products and services to a global audience of customers. Web storefronts are limited to one seller, i.e., they enable a seller to use the Internet to differentiate its product offerings; enhance customer service; and lower marketing, sales, and order processing costs. For example, a shoe store can develop a Web Storefront that allows customers to purchase shoes over the Internet. As shown in Figure 12-5, storefronts support Web based purchasing systems that allow users to search company catalogs for certain price ranges and then place orders for chosen product(s). This represents online buying/selling through a catalog using a shopping cart, electronic wallet, or similar tool. It includes both consumers purchasing goods and online buyers purchasing goods from a supplier. It can also include links to back-end systems for inventory updates and credit checking.

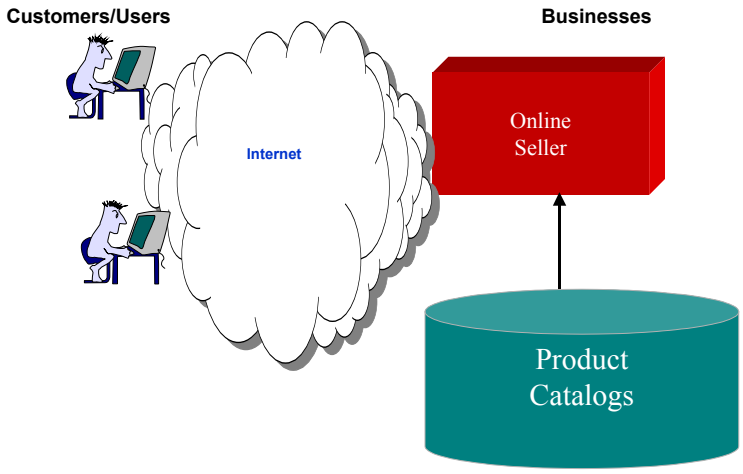


Figure 12-5: Online Purchasing Through a Storefront

A very large number of Web storefronts currently exist. Examples are:

- Staples.com -- for buying office supplies online
- E-Bay – for buying numerous products
- Shop.com -- for buying groceries
- Flowers.com – for buying flowers

Storefronts basically show a company's presence on the Web and are usually based on a product catalog that shows product features, price, expected delivery time, etc. These web-based sales solutions deliver process and cost improvements to sellers but they are very "supplier centric". These supplier-centric solutions can complicate efforts of

customers to control expenditures and maintain preferred supplier relationships. For example, you may have to visit several storefronts to find a bargain.

Virtual Shops. Virtual shops go a step beyond the Web storefronts by providing a storefront that represents several back-end sellers. In other words, the restriction of a single seller is removed. For example, Amazon.com supports purchase of books by tying several bookstores together. Enterprises that support virtual operations are known as "virtual enterprises" or extended enterprises. Basically, a ***Virtual Enterprise (VE)*** is a network or loose coalition of a variety of value adding services in a supply chain, that unite for a specific period of time for a specific business objective, and disband when the goal is achieved. Examples of virtual enterprises, in addition to Amazon.com, are:

- Drugstore.com -- To buy drugs online (many partners)
- Virtual Parts Supply Base (VPSB, http://www.vpsb.com/) to supply hard to find parts for US Government.
- The National Industrial Information Infrastructure Protocols (NIIIP) Consortium to develop inter-operation protocols for manufacturers and their suppliers (for more information on NIIIP see http://www.niiip.org).

Virtual enterprises can be, if needed, customized to reflect a buying organization's unique trading agreements, workflow, and business rules. These virtual procurement channels, also known as *e-procurement*, enable a self-service purchasing environment that pushes product selection and order initiation to the desktops of frontline employees through a common Web browser. Many early e-Procurement solutions were intranet-based applications that did not fully leverage the ubiquity of the Internet. As a result, many e-Procurement solutions are now transitioning to e-markets. Many unique issues in virtual shops and virtual enterprise arise.

Online purchasing systems, as stated previously, involve a large number of components such as Web pages, search engines, product catalogs, order processing, payment, inventory control, and shipping/receiving systems. Figure 12-6 shows the various components of an online purchasing system. These components may reside on different computers and may be built by using different technologies. In addition, all these components may need to be configured differently and will need to work together in a secure fashion to satisfy the demands of online traders.

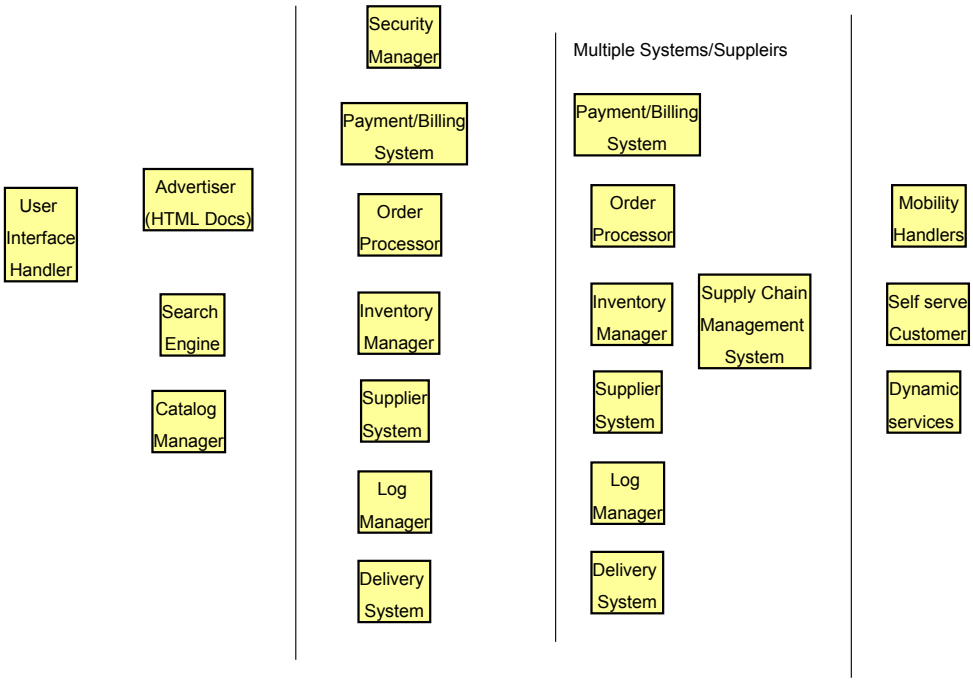


Figure 12-6: Logical Components of Online Purchasing

12.3.2 Online Purchasing Security Through SET

Many issues related to basic e-commerce (online purchasing) have been discussed previously in the general distributed applications security. In other words, the clients, the servers, and the back-end systems need to be secure in addition to the underlying platforms and networks. The main thing that has not been discussed is secure electronic payments -- how do you make sure that the credit card information is secure and private during the purchase process.

Many credit card payment systems use the secure sockets layer (SSL) protocol for encrypting the credit card payment data. However, SSL does not verify that the purchaser is the owner of the card being used for payment. SET **(**Secure Electronic Transaction) was developed jointly by Visa, MasterCard, IBM, and other technology providers for secure credit card processing. SET is used to protect the transfer of bankcard payment information over open networks like the Internet. This is an application layer security protocol that is used primarily for credit card processing. Unlike SSL that encrypts all communications between a client and server using TCP/IP, SET is highly specific to credit card processing and contains logic that is based on the dance ("choreography") between four players: consumer, merchant, merchant bank, and consumer bank.

Figure 12-7 illustrates how SET works in credit card based purchasing. Before starting, a user acquires a digital certificate and SET-enabled digital wallet. The wallet and certificate contain the identity of the user and the credit card being used. This verifies that

the purchaser is the owner of the card being used for payment. In the first step, the user shops at a Web site that uses the SET payment method. As part of this step, the merchant's servers send a message over the Internet to invoke the user's SET wallet. The digital wallet encrypts the payment information and sends it to the merchant. In the second step, the merchant encrypts this information and passes it on to the merchant bank. The merchant's bank sends this encrypted information to the customer's issuing bank in the third step. In the fourth step, the customer bank approves or denies the transaction based on credit standing. The CB also performs credit authorization before approving (i.e., does this payment go over the consumer credit limit). In the next few steps, the transaction is completed. If the payment is authorized, the merchant's bank arranges for the fund transfer from user to merchant and the user's credit card account is charged for the transaction amount. The merchant then ships the merchandise to the purchaser.

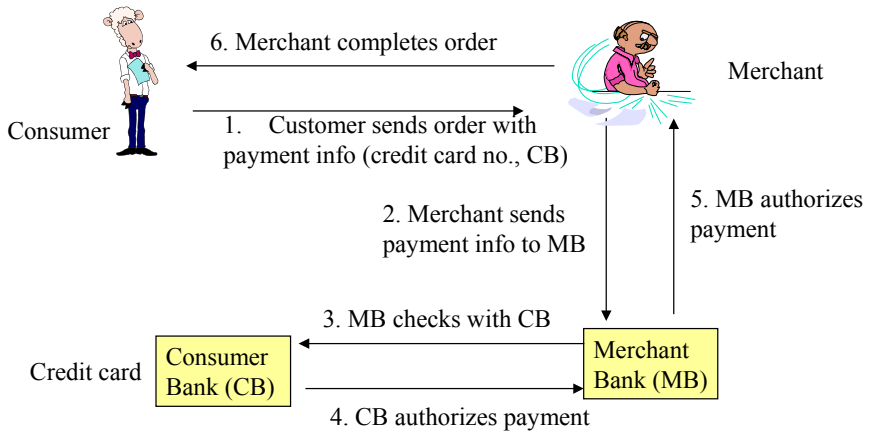


Figure 12-7: SET Processing

Although the actual SET processing is quite intricate, from an end-user's privacy point of view, the purchase transaction is separated in two parts by SET:

- Purchase information that is sent to the merchant
- Credit card information that is only handled for credit card verification by the CB. This information is encrypted so that the merchant does not see and cannot store the credit card information.

The consumer's credit card information is thus kept private even from the merchant by SET – the merchant only sees what goods he is selling, how much money will he be paid, and if the credit was authorized. This level of processing cannot be done by SSL, i.e., SSL cannot be used to check credit a card for validity, checking for credit authorization, and processing the payment transaction. An organization called SETCo manages the SET specification and promotes/supports the use of SET Secure Electronic Transaction on the Internet. See their site (www.setco.org) for additional information about SET.

While SET is quite popular, other payment systems such as cybercash (www.cybercash.com) and DigiCash (www.digicash.com) are also available.

12.3.3 Weak Server-Side Programs – The main Weakness of e-Commerce Security

Figure 12-8 shows a functional view of e-commerce -- the main activities that are performed in purchasing. Most of these functions are implemented through server-side programs that authenticate the user, check available items, facilitate buying/selling, and handle payments. The main goal of e-commerce attackers is to gain control of these programs, so let us look at a potential scenario.

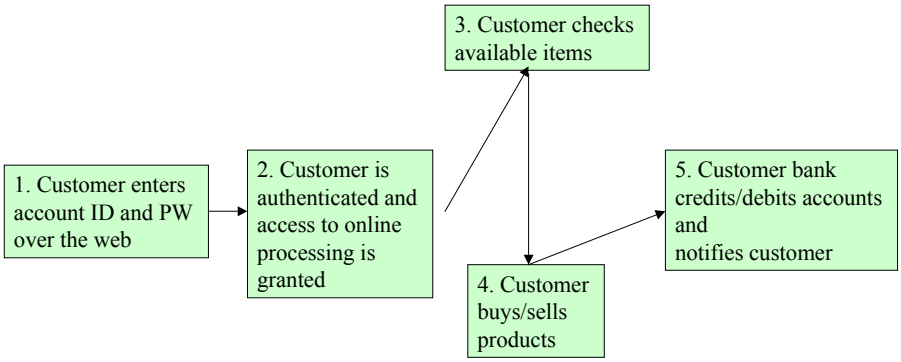


Figure 12-8: Functional-View of e-Commerce

Figure 12-9 shows how the e-commerce systems are typically implemented. This is, naturally, one out of many possible configurations. In these systems, server-side programs receive the requests from the browsers, then access databases and then invoke back-end systems. The following scenario is common [Ghosh 2001][1]:

- The attacker uses a Web browser over the Internet to invoke weak server side programs (CGI scripts, servlets). These weak programs could be test programs that were left on the system inadvertently, a default CGI scripts from an application server, or a program with faulty logic that was never debugged and corrected. Many server side programs do not check inputs rigorously. This weakness can be exploited by the attackers to pass unexpected inputs and gain privileged status.
- The goal of most attackers is to spoof the web server programs to gain privileged status because once in this status, the attacker can access almost any file and invoke other programs. The attacker can possibly read the directories with user IDs and PWs and can possibly gain control of the system.
- A typical trick by the attacker is to install a trojan horse or a backdoor by taking advantage of the priviliged mode. These backdoors/trojan horses can be used by the attackers later to enter the system unnoticed.

What can be done? The best way to defend against such attacks is to "harden" the server-side programs. Specifically, do not to leave untested and flawed code on the machines. In particular the "test" server side programs need to be carefully watched because they can be invoked by users from the Web browsers anywhere on the Internet to enter your system.

[1] Ghosh, A., "Security and Privacy for e-Business", Wiley, 2001

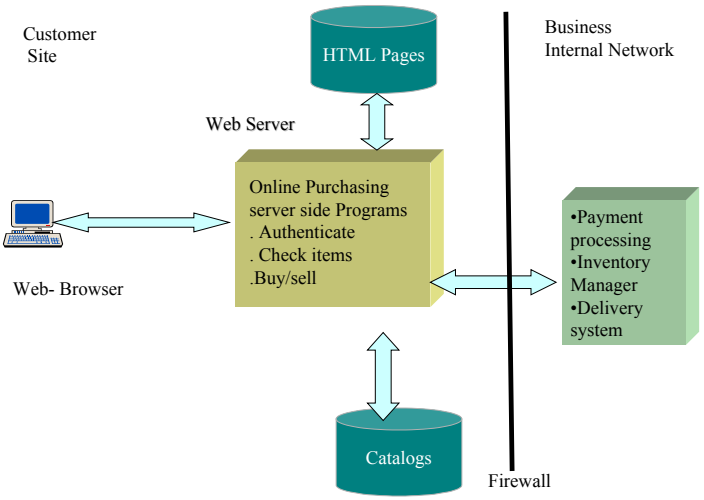


Figure 12-9: On-Line Purchasing

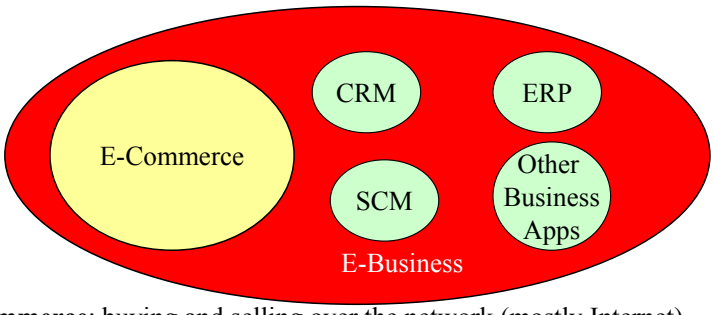
12.4 e-Business Applications Security

12.4.1 Overview

Simply stated, *e-business (EB)* is conducting business, including buying/selling, over the network (mostly Internet). Thus, EB = EC + other activities such as conducting meetings, developing software, and managing customer relationships. In this sense, EB subsumes EC (see Figure 12-10). The main idea is that e-business is concerned with delivering business services to its customers as well as its partners, electronically. Thus we will stay with our simple definition, (i.e., "e-business is conducting business, including buying/selling, over the network (mostly Internet)"). For example, use of the Internet to conduct meetings, develop and deliver software, provide library services, and support customer relationship management systems is all considered as part of e-business -- plus, of course, any of the traditional e-commerce activities of buying/selling over the Internet (see Figure 12-10). Since e-business subsumes e-commerce, we will use e-business (EB) to discuss other applications of e-commerce (EC) that were not discussed previously.

To discuss the e-business applications that need to be protected, we will use Figure 12-11 as a basis for classifying e-business applications:

- Business to consumer (C2B) applications such as online buying and web publishing.
- Business to employee (B2E) internal business applications such as enterprise resource planning (ERP).
- Business to business (B2B) trade conducted between partners directly (e.g., supply chains) or through an intermediary (e.g., an emarket).
- Other players represent consumer to consumer (e.g., collaborative computing) and consumer to data (e.g., e-business intelligence) applications.



E-Commerce: buying and selling over the network (mostly Internet)

- Advertising
- Browsing/selection
- Purchasing
- Payment/payment settlement

E-Business= E-Commerce + Other business activities conducted electronically

- Conducting meetings for geographically distributed parties
- Software development and delivery across multiple organizations
- Delivering on-line instruction and training
- Customer relationship management (CRM)
- Supply chain management (SCM)
- Enterprise resource planning (ERP)
- Knowledge management

Figure 12-10: e-Commerce Versus e-Business

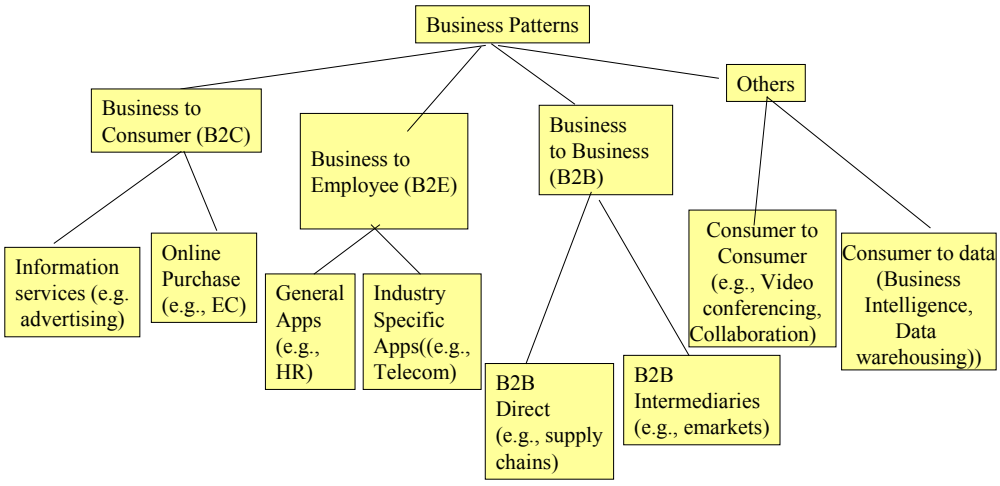


Figure 12-11: Classification of e-Business Applications

12.4.2 Consumer to Business (C2B) Applications Security

Consumer to Business (C2B) applications include the general case of internal and external users interacting with enterprise transactions and data. These activities are

particularly relevant to enterprises dealing with goods and services that can be listed and sold from an online catalog. It covers two parts: consumer information services and consumer online purchasing.

12.4.2.1 C2B Informational Services (Web Advertising and Customer Relationship Management)

This essentially covers user-to-business interactions that do not involve any purchasing. Example are web-based advertising ("web publishing") and customer relationship management (CRM). Figure 12-12 shows a simple C2B information application.

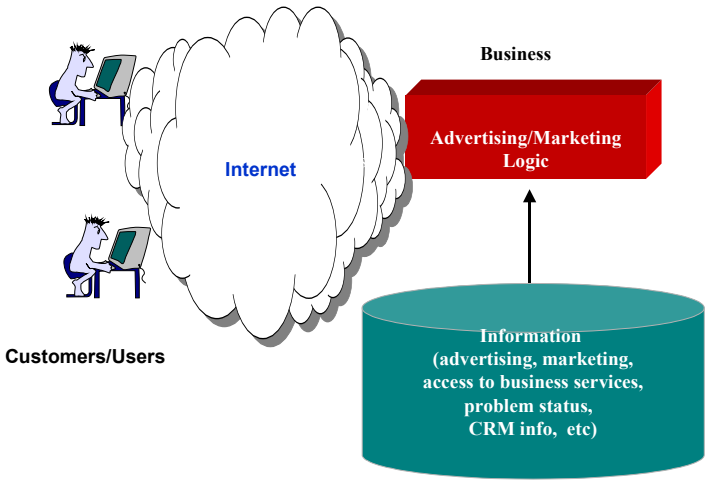


Figure 12-12: C2B Information Services

Web advertising is one of the oldest applications of the Internet that became popular in the mid 1990s. This type of application is the foundation of many corporate web sites. The web sites are used to display/advertise company products and services. The customers have to separately order the products that they select by browsing through company Web sites.

Another example is Image processing applications. Specific examples of these applications include sending and processing of images such as X-rays in the medical industry, photographs used in claims processing for the insurance industry, proofs and advertisements in the publication industry, and visualization of systems dynamics in the aerospace industry. These images may be sent between users to businesses as well as business to business entities.

Customer relationship management (CRM) enables organizations to identify, attract and increase retention of profitable customers, by managing relationship with them. CRM systems are an outgrowth of the traditional customer care systems that concentrated on customer loyalty through improved service and communication. At present, CRM has evolved into a collection of methodologies, software, and Internet capabilities that help an enterprise manage customer relationships in an organized way. CRM applications -- often used in combination with call centers, data warehousing, and E-commerce applications -- allow companies to gather and access information about customers' buying histories, preferences, complaints, and other data so they can better anticipate what customers want and need.

Security Issues and Approaches. Because these applications are informational only, the security challenges are somewhat muted. The unique security issues of C2B information services applications with corresponding suggested approaches are:

- Privacy issues could be raised if the consumers do not want the web sites and other resources they visit private. SSL does not help here because it cannot encrypt the URL of the site that you invoked. A proxy server that "anonymizes" the web sites may be needed.
- Integrity issues should not be raised because the information is retrieval only.
- Authorization and authentication issues may arise especially for CRM systems. Most commercial CRM packages at present support a variety of authorization and authentication schemes.
- Accountability issues may arise but can be addressed by logs and audit trails maintained by the Web sites.
- Availability issues arise routinely due to the denial of service attacks on web sites. Replicated Web sites could be used to circumvent this threat.

12.4.2.2 C2B Online Purchasing - e-Commerce

This application represents online buying/selling through a catalog using a shopping cart, electronic wallet, or similar tools. We have already discussed this in a previous ection under basic e-commerce security.

12.4.3 B2E Internal Business Applications – The ERPs (Enterprise Resource Planning)

The business activities, such as flow of purchase orders between various business units of an organization, require linking together of applications within a business. A common example is Enterprise Resource Planning (ERP) systems that support back-office operations. ERPs support inventory management, order processing, and financial reporting applications. Traditionally, enterprise resources have been managed by a multitude of independent applications in human resources, payroll, order processing, inventory control, billing, and accounts payable/receivable systems. The basic idea of modern ERP systems is that they provide an integrated database approach to manage and operate enterprise resources such as employees, materials, and services.

ERP is not one application – it is a collection of applications that can be further categorized in terms of the enterprise resources they manage (see Figure 12-13):

- The basic ERPs manage the core resources that are common to all organizations. Examples of these resources are people, costs and assets (e.g., building, furniture, etc). Several ERP systems are designed to manage these resources. ERP systems of this type are available from suppliers such as Peoplesoft, SAP, and Oracle.
- "Vertical" ERPs concentrate on managing resources in specific industry segments. The oldest examples of ERPs in this segment are the ones that manage manufacturing resources such as materials, finished goods, bill of materials (i.e., the materials used in building a finished product), and inventories. These ERPs also integrate order processing applications with manufacturing materials and inventory management systems for integrated operations. SAP has developed an extensive suite of ERP applications for manufacturing. Another area of ERPs in vertical markets is the ERPs in telecom markets that manage telecom resources. These ERPs

are known as Operation Support Systems (OSSs) in the telecom marketplace. At present, many OSSs are homegrown and developed by the telcos. However, established ERP vendors such as SAP are also beginning to provide OSSs

In general, the current trend in ERPs is to manage *all* enterprise resources that include the core, vertical, and service resources in an integrated manner. This includes human resource, general ledger, payroll, order processing, inventory control, and other "classical" business applications.

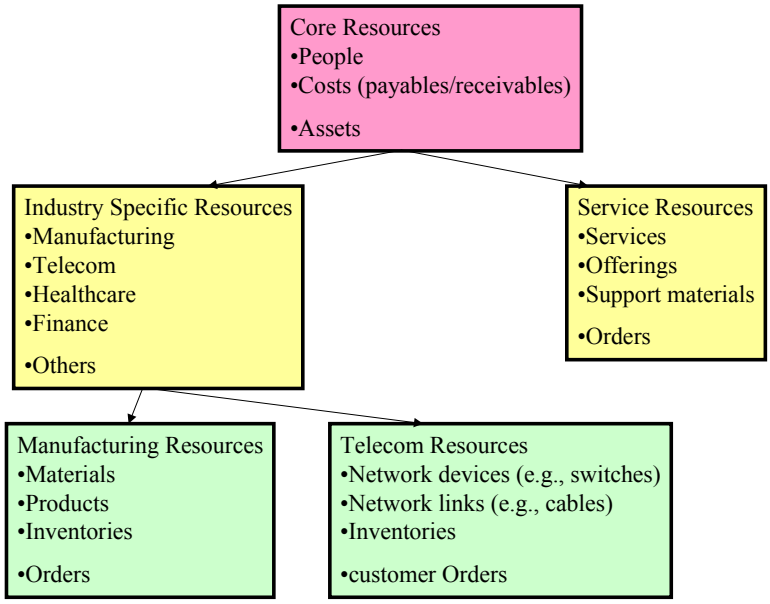


Figure 12-13: ERP Categories

Security Issues and Approaches. From security point of view, ERP systems present several challenges. Some of these applications are legacies, some are packages provided by vendors such as SAP and Peoplesoft, and many are mixtures of everything. For legacy applications, the issues are somewhat similar to the legacy application security issues discussed previously. For packaged applications, you need to understand the security features of the packaged applications and assure that they satisfy your security requirements. The main challenge is the mixture where many applications from many suppliers have been integrated (usually patched) together. This is a nightmare from a security point of view because many of these systems have different security features that are incompatible with each other. However, there are a few areas of comfort:

- The ERP systems are typically back-office applications that run in a protected corporate Intranet. In many cases, these applications reside on one mainframe system that can be monitored and controlled relatively easily.
- ERP systems usually contain information that is typically boring to the average hacker. This does not exclude industrial spies who want to steal corporate secrets such as customer information, product plans, and the like.
- Because of their mainframe “culture”, the skills to break into ERP systems are not commonly found in the hacker community. Typical hackers know a great deal of Unix kernels and Microsoft Windows environments but know little about IBM mainframe systems.

The most pragmatic approach to secure ERP systems consists of the following steps:

- Make sure that the lower level issues of firewalls, networks, and platforms have been properly handled to provide a secure environment for these systems.
- Separate the systems into packaged applications and home grown applications.
- For packaged applications, do the best you can to secure the system by using the security features provided by the package suppliers.
- For home grown applications, do your best to use the existing features and add others if needed.
- Carefully examine and secure the paths (the links) between the various applications (packaged, home grown). This is essential, because even if the individual applications are secure, their communications with each other may not be secure.

12.4.4 Business-to-Business (B2B): Supply Chain and eMarket Security

Business-to-business (B2B) activities (purchasing plus informational) are vital to e-business and thus important for security. These activities fall into two broad categories: a) B2B direct where the business activities are conducted directly between trading partners and b) B2B indirect in which the trade partners use emarkets as intermediaries,

12.4.4.1 Business-to-Business Direct (Supply Chains)

This characterizes business activities directly between trading partners -- it is assumed that an agreement between the trading partners exists. An example is supply chain management between suppliers of parts and corporations. The main idea is that interactions between partners form a shared process, or potentially multiple distinct shared processes (Figure 12-14). Partners need to retain organizational independence, and so processes are categorized as either public or private. Public processes are shared between businesses and private processes are not shared -- they are fully under the control of the individual partners. A good example of business-to-business direct is supply chain execution in which automated processes work across a supplier network. Supply Chain Management (SCM) is an enterprise wide infrastructure for managing a network of facilities and distribution options such as procurement of materials, transformation of these materials into intermediate and finished products, and distribution of finished products to customers.

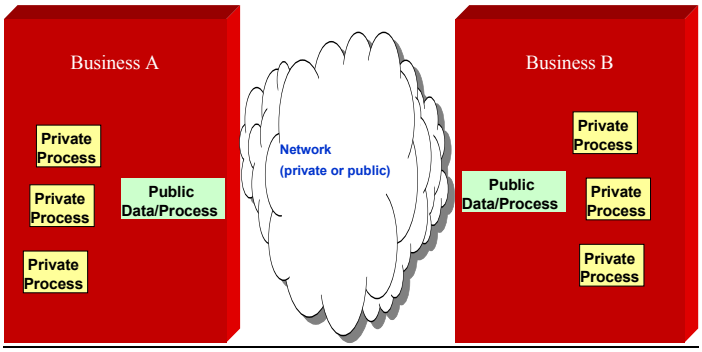


Figure 12-14: B2B Direct (Supply Chains)

There are several successful examples of multi-national corporations that reap benefits and maintain their competitive edge due to efficient SCMs. Some representatives of industry practitioners include Procter & Gamble, Wal-Mart, Coca-Cola, Hewlett Packard, Cisco, IBM, Sun Micro Systems, Compaq Computers, Dell and 3COM. The market for SCM is growing due to mergers among corporations, new e-commerce virtual enterprises, change/expansion in company focus, new customer demands and global competition.

Security Issues and Approaches. There numerous security issues related to supply chains. A Most of the issues are concerned with the transmission of information between the players and the public versus private data/processes.

- For privacy and integrity, the paths between the players must be properly secured. This can be done by using VPN or a jointly owned extranet. In addition, the public processes and data must be carefully specified to assure that unnecessary information is not exposed.
- For proper authorization and authentication, appropriate mechanisms should be used at the public interfaces. For example, if the public processes are Web-based, then SSL could be used. In particular, the public interfaces should implement strong authentication and authorization features (digital signatures and certificates) for highly sensitive information exchanges.
- For accountability, a clear and concise log of all activities related to supply chains must be maintained.
- For availability, it is important to maintain a reliable network (perhaps an extranet, if needed) and a public process that runs on a reliable machine.

12.4.4.2 Business-to-Business Indirect (eMarkets)

In this case, the trade partners use emarkets as intermediaries. The emarkets support multiple buyers and suppliers for auctions, reverse-auctions, and brokerages. B2B operations through intermediaries such as emarkets bring together buyers and sellers to provide efficient electronic trading of goods and services. An emarketplace is an electronic gathering place that brings together multiple buyers and sellers. An emarketplace provides its members with a unified view of goods and services and lets its members perform transactions electronically (see Figure 12-15).

Emarkets bring together multiple vendors "under one roof" and provide a single point of access for brokering financial transactions and information exchange across a large community of buyers and sellers. They offer a powerful means for purchasing based on vendors, price, terms, order, payment plans, etc. The participation of diverse suppliers and auctions/reverse auctions differentiate emarkets from Web-storefronts and virtual shops. A large number of emarkets (EMs) are being developed at present. An interesting example of emarkets is www.Verticalnet.com that provides several marketplaces in a diverse array of segments. Current EMs are beginning to consolidate a wide range of intermediaries such as the following:

- Clearing houses that provide a common point for traders (e.g., catalog clearing houses)
- Trading hubs that allow customers to trade goods and services (e.g., BandX for bandwidth trading)
- Brokerages that provide brokers that act on your behalf (e.g., shopbots that shop on your behalf for certain items based on your needs)

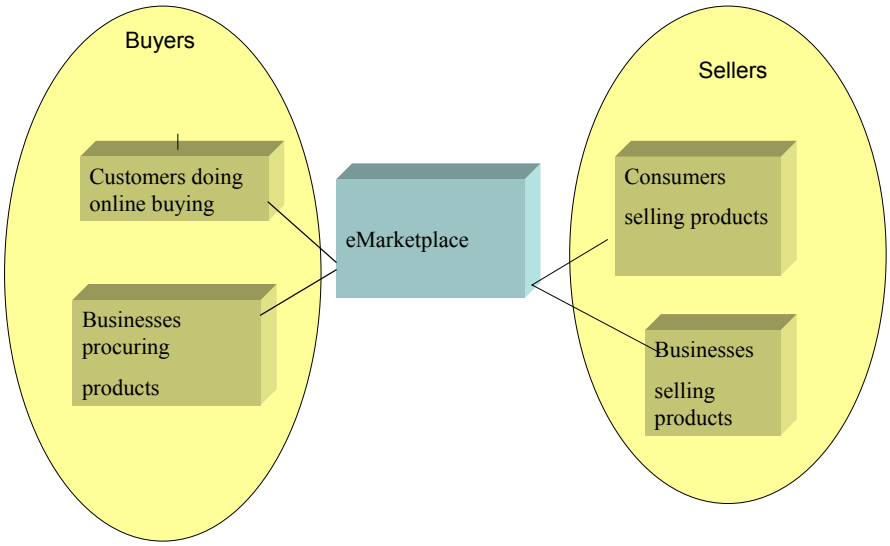


Figure 12-15: B2B – eMarkets

Security Issues and Approaches. B2B emarkets also raise numerous security issues. Many of these issues are similar to the supply chains because both involve B2B trade. Thus the previous discussion of transmission of information between the players and the public versus private data/processes for supply chains also applies here. However, there are several unique issues also:

- Emarkets involve purchasing, thus money transfers, credit approvals, etc arise. Standrads such as SET can be used to assure privacy and security of online purchasing.
- Most current and future emarkets use XML extensively. For example, the purchase orders and item descriptions are largely XML-based. In addition, emarkets use XML repositories that contain XML DTDs or schemas for common items. Security techniques such as XML encryption and XML signatures may be of value here.
- For Web Services based emarkets, SAML may be very beneficial because it can be used standardize security assertions between multitude of partners.
- For international emarkets, numerous international exchanges, legal, and regulatory issues arise.

12.4.5 Consumer to Data (C2D) Applications – Business Intelligence

In these applications, the consumers directly interact with the databases for business intelligence. A specific example is data warehouses that are used to support business intelligence through data mining and other processes. Data warehouses have been established in many organizations to provide access to operational data by creating a repository for decision support. Data mining tools are increasingly playing a key role in data warehousing because they utilize statistical analysis and pattern recognition techniques to answer business questions. These tools exploit a combination of AI and statistical analysis to discover information that is hidden or not apparent through typical

query and analysis tools. The availability of massive amount of corporate data in data warehouses has provided a rich field for data mining. Data warehouses -- typically large relational databases-- are widely accessible from Web. The users of the warehouses employ Web browser based tools for querying and analysis. Figure 12-16 shows a conceptual view of data warehouses.

Security Issues and Approaches. The main security issues related to data warehouses include protection of the data warehouse itself and then authentication/authorization of the users. Most of these issues are not unique and can be dealt with the same type of security measures needed for Web to data-store applications (e.g., secure the path and authenticate/authorize the users strongly).

12.4.6 Consumer to Consumer (C2C) Applications - Collaborative Computing and Groupware

These applications allow consumers to directly interact with each other. Specifically, these applications include computer supported cooperative work (CSCW) that allows people and processes in different parts of an organization to work together. Within this somewhat broad and vague umbrella, many application areas ranging from computer assisted instruction to business process automation have emerged. Some areas such as groupware, collaborative learning, and workflows have especially gained momentum. Let us talk about a few briefly.

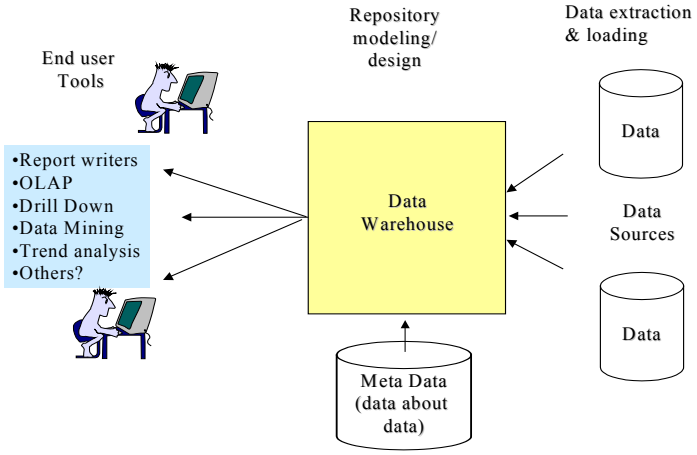


Figure 12-16: Data Warehouse

Groupware systems allow users to interact with each other by supporting document preparation and file/email exchanges. More recent systems use Web browsers to exchange voice, data and video for numerous office applications. For example, new teleconferencing systems display text, pictures, and video/voice on different windows of a workstation. Collaborative computing applications go beyond the computer conferencing and groupware software to cooperatively solve problems. These systems may include high definition TV and "artificial life" animations. An example is the extensive "decision support systems" which allow location independent teams to work cooperatively as if they were in the same room (talk, see each other, review each other's

documents, etc.). It is important to provide all these facilities through one common user interface (i.e., the Web browser).

An example of groupware is IBM's Lotus Notes. It consists of one or more Notes Servers that are connected to the Notes Clients over an enterprise network (see Figure 12-17). The Notes Server houses the Notes building blocks: databases (collection of Notes documents), documents (text and graphics files), views (list of documents), forms (description of document structure), and fields (predefined fields in a form). The Notes Server manages the repository of information for the workgroup members.

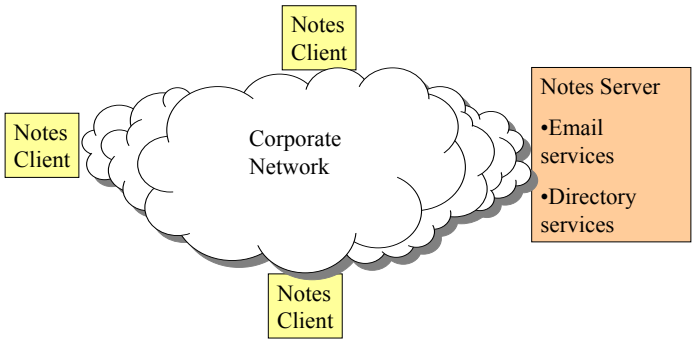


Figure 12-17: Conceptual View of Lotus Notes

Another example is Communities of Interest (COIs) which allow users from the same community to exchange messages, transfer files, send mail and exchange XML documents to conduct business over the Internet. For example, the real estate COI users access real estate documents and databases through Web browsers. COIs have become more sophisticated with incorporation of workflows and multimedia presentations.

Very interesting collaborative learning and multimedia applications are being developed for higher education by the NSF (National Science Foundation) funded **Internet2** project (www.internet2.org). Examples of these applications include virtual laboratories, digital libraries, and distributed instruction (see the Internet2 web site for more details).

Security Issues and Approaches. Collaborative applications may need sound security measures if the collaboration is on a sensitive topics. It is a good idea to authenticate and authorize the participants and protect the shared documents. Most groupware and collaborative application platform provide their own security. For example, Lotus Notes has its own security solution.

12.5 Mobile Application and Mobile Commerce Security

12.5.1 Overview

Mobile applications allow hand-held devices to access and use a wide range of databases and applications for e-banking, retail payment, brokerage, and e-business. Examples of the mobile applications are:

- *Mobile* Enterprise Business Applications (MEBAs) that add the mobility dimension to EB applications such as ERPs, SCMs, CRMs, etc. This allows employees, partners, and customers to use mobile devices such as laptop computers, personal digital assistants (PDAs), and digital telephones to conduct business.
- Mobile Commerce (M-Commerce) applications that allow cellular phones and PDAs to search the Internet, access data and information, and conduct purchasing or business transactions.
- Voice Commerce (V-Commerce) is gaining importance to support users who want to use telephones and other voice-driven devices for conducting e-commerce. This includes "Voice Portals").
- Positional Commerce *(P-Commerce)* is becoming popular to provide support to the customers based on their geographic position (e.g., give you information about deals in the Atlanta area when you are in Atlanta). The systems use a GPS (Geographical Positional System) to locate the position of the customers.

12.5.2 Mobile Applications Issues

Many of these applications raise security and privacy concerns. In particular, online transactions from handsets can result in fraud and theft with huge financial losses. In addition, the positional commerce raises some privacy issues (you may not want everyone to know that you are visiting Atlanta). Additional concerns about mobile users (e.g., increased chance of eavesdropping) also exist.

In general mobile application security involves the following issues:

- Wireless network security issues
- Wireless platform security issues
- Mobile application security issues that focus on the business aware code written for mobile applications.

We have already discussed the wireless network and wireless platform issues previously. Let us now concentrate on the mobile application specific issues. Specifically, let us review Figure 12-18 for a conceptual understanding of mobile application security risks and possible approaches. The main difference is at the front-end (i.e., wireless network and the front-end integration layer). Thus the network and the front-end layer must be able to handle all mobile security concerns. We have already discussed the wireless network issues in a previous section, so let us focus on the front-end layer that must support mobility specific software in a secure manner.

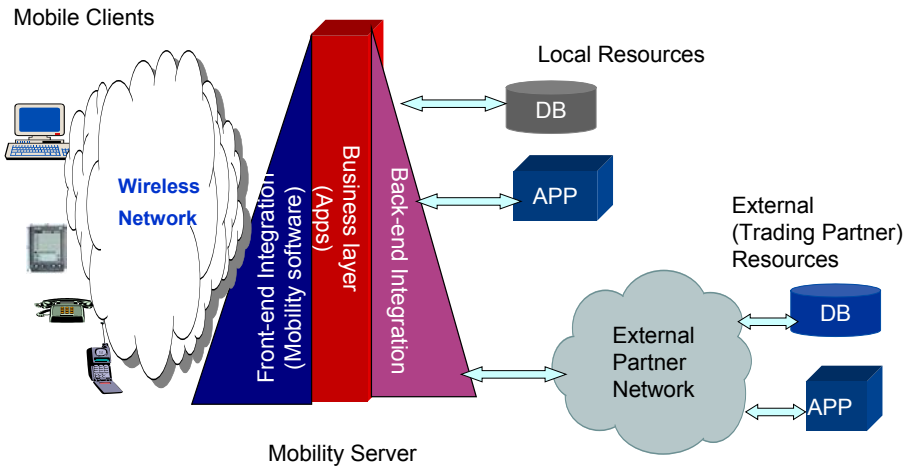


Figure 12-18: Conceptual Architecture for Mobile Applications

The mobility specific software is responsible for "roaming support" (e.g., the GIS Map for GPS), voice support for voice commerce, and uniform access to the CRM, ERP and proprietary or custom-developed business/commerce applications. The security features vary widely. For example, if WAP was used then WAP security discussed previously could be used but if other technologies such as I-Mode are used that lack security support, then the deficiencies need to be compensated for. The best approach in this situation is to build a separate "mobility" proxy that performs the front-end integration task and also provides the necessary security checks for integrity, privacy, and other requirements. The mobility proxy could also serve as a firewall (an application level gateway) that verifies and filters the traffic before attaching to the back-end systems.

The mobile application security can be discussed in terms of mobile client security, web tier security, and the back-end transaction security. However, most uniques issues are concerned with the mobile clients -- the web tier and back-ends are not mobile and thus do not need additional discussion.

12.5.3 Mobile Client Security

Mobile client software residing on hand-held devices is responsible for identifying the user with passwords, or biometrics such as voice recognition. In addition, the encryption/decryption software as well as digital certificates, if employed, resides on the handheld devices and is the responsibility of client software. Typical threats on client side security are based on "active content" that appears as:

- Helper applications and plug-ins that are used to handle special programs (e.g., Powerpoint) that are not shipped with web browsers
- Java script and VB script programs that are used to imbed executable code in HTML pages
- XML processing on the client side that uses Extensible Stylesheet Language (XSL) to convert XML to HTML
- Java applets that can do a variety of tasks on the browser (e.g., show video clips, draw graphs and charts)
- ActiveX control applets that control the display of MS Desktop Services

- XML/SOAP clients that access remote resources in the Web services and MS .NET environments
- Browser-side cookies that keep track of the activities at browser sites

While active content on client side poses some security risks in all environments, it can be particularly serious in mobile devices. Mobile, especially handheld devices, can be stolen and any of these programs can be compromised, replaced, and/or modified by the intruders. Due to space limitations, it is beyond the scope of this chapter to discuss all these topics in detail. We mention a few for completeness.

Client certificates can be required for client side authentication. However, these certificates pose special management problems because the key pair associated with the client certificate resides only on the client. Many systems such as WAP and I-mode produce certificates that are stored on the clients (PDAs, cellular phones). The main problem is that the client certificate, once stored on the handset, has to be safeguarded and managed until the certificate expires. This creates several additional problems because the client certificates go with the device if the device is stolen. Another problem is that the key pairs must be generated on the mobile device. These certificates can also be generated and loaded onto the mobile devices. Instead of certificates on mobile devices, the client may refer the wireless gateway to a directory to retrieve the client certificate from a directory. This saves the communication bandwidth needed to send the client certificate over the air, however, the wireless gateway must trust the directory the client refers to in order to assure authentication. The certificate directory also must be available at all times to allow users to retrieve the certificate when requested.

Java security is an important aspect of client side protection for mobile devices. Security of Java code has been an area of concern for a while and is important for handheld devices because some phones, such as I-mode phones, include Java code. In addition, wireless Java, part of the Sun J2ME (Java 2 Micro Edition) is used in many handheld devices. The current Java security is defined at the following levels:

- Java1.1's security management system. All local code is trusted. All remote code is un-trusted, unless it is digitally signed by a trusted source. Un-trusted code runs in a "sandbox", and has limited access to local system resources.
- Java 2's security management system. Local and remote code are checked by the same security management system. It enforces fine-grained, flexible and easy-to-specify security and permission policies.

12.6 Mobile Agent Security

12.6.1 Overview

Mobile agents represent a very different aspect of mobile applications. In e-commerce, mobile agents can roam around the network looking for information and bargains on behalf of the customers. Most mobile agents are Java applets that go from one computing system to another. An important area of concern is the mobile agents because they move from site to site and thus introduce numerous security risks of viruses, time bombs, Trojan Horses, and the like discussed previously. Due to these concerns, the use of

mobile agents in enterprise computing is greatly restricted. However, this is an interesting area of development that needs a closer look.

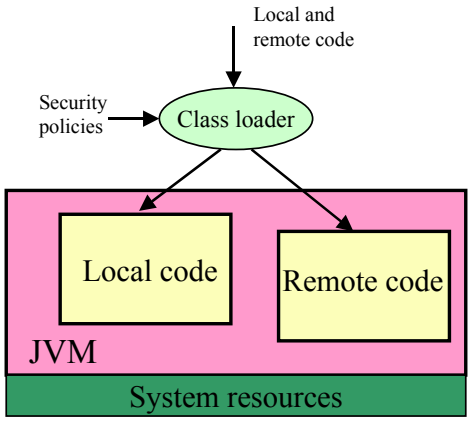


Figure 12-19: Java Security Model

What are mobile agents? Let us first start with agents. An ***agent*** is a software entity (i.e., a program) that has some degree of autonomy. It carries out operations on the behalf of a user or another program, and in this process, represents or has knowledge of the user's goals and wishes. In this sense, a software agent is similar to a real life agent such as a life insurance agent, a car insurance agent, a travel agent, a real estate agent, and the like. All agents, software or human, carry out a set of operations on behalf of a user (customer) -- they do so with some degree of autonomy to satisfy its user's goals.

Software agents, or real life agents, can be:

- Intelligent or dumb (I am thinking of my life insurance agent)
- Static or mobile

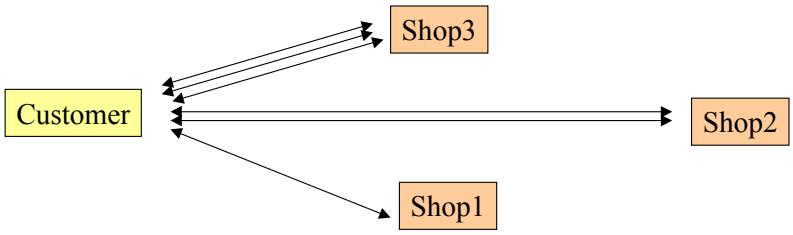
Intelligent software agents (henceforth referred to as ***intelligent agents)***, have the ability to learn, use knowledge effectively, and adapt to new situations. An example of intelligent agents is a ***"shopbot"*** that shops on your behalf, looking for bargains and brokering to get you the best deals. An intelligent agent is typically;

- *Knowledgeable:* has knowledge of a domain (e.g., insurance) and user needs
- *Self-learning:* can acquire additional knowledge from different situations
- *Pro-active*: takes initiative; sets and pursues goals
- *Autonomous*: decides what to do without external human intervention
- *Timely*: does not spend forever deciding what to do next
- *Persistent*: remembers a "lifetime" of activity
- *Social and communicative*: interacts with other agents

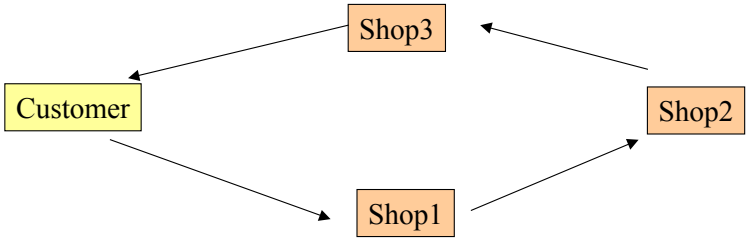
Mobile agents are programs capable of being transferred to remote hosts in order to carry out different tasks on behalf of their users. Mobile (transportable) agents have the ability to travel through the network. A Mobile agent can halt its execution, move to another host on the network while maintaining its state, and resume execution on the destination host. Mobile software agents are also similar to mobile real life agents who travel around on your behalf instead of sitting around and making phone calls or sending email. I can , for example, ask my nephew to buy a lawn mower for me by driving around in the neighborhood instead of making phone calls and getting on web sites.

Although mobile agents can be dumb, we will assume that the mobile agents of interest to us are intelligent. Thus we can think of a mobile shopbot that moves around a trading network to shop on our behalf by using certain level of intelligence in looking for bargains.

Mobile agents basically provide an alternative to the very common client/server model. In a client/server model, the clients send the data to the program sites (servers) and receive the results. In a mobile agent model, the program is shipped to the data sources -- it thus travels from one data source to the next, collects the results, and sends the results back to the originator. Figure 12-20 illustrates the differences between client/server and mobile agents by using a shopping example. Suppose you wanted to find a cheap computer quickly. In the client/server model, the customer issues calls to different shop sites, in some cases multiple calls are issued to the same shop. The results are sent back to the customer after every visit to a shop site. In a mobile agent model, a "shopbot" is sent to the first shop where it checks for the desired computer. It then moves to the next shop (carrying the results from shop1). After shop2, the shopbot moves to shop3, carrying the results from shop1 and 2. The accumulated results are sent back to the customer after shop3.



a) Shopping by using a Client/Server Model



b) Shopping by using a Mobile Agent Model

Figure 12-20: Mobile Agent Versus Client/Server Model

Extensive lists of agent technologies (mobile as well as static) can be found at the web sites www.informatik.uni-stuttgart.de and www.reticular.com.

12.6.2 Sample Applications of Mobile Agents in Ecommerce

Although the mobile agent technologies have been around for a while, real life applications are relatively sparse. We have already discussed shopbots that are mobile intelligent agents which go around and shop on your behalf. Let us consider a more detailed example of this.

A company that needs to order office supplies could use agents ("inventory agents") to monitor the quantity and usage patterns of office supplies within the company and launch buying agents when supplies are low. The buying agents can roam around the network, automatically collecting information on suppliers and products that fit the company needs. They can also decide which suppliers and products to investigate in detail, negotiate the terms of transactions with selected merchants, and finally place orders and make automated payments. In this example, the inventory agents may or may not be mobile but the buying agents should be mobile.

Similar applications of mobile agents in Ecommerce are:

- Personal agents to go around the network to collect and present information to you in the way you want it (e.g., sort the sites you want to visit in terms of historical significance)
- Mobile automated negotiators for retail e-commerce, bandwidth trading, subcontracting for manufacturing, electronic trading of financial instruments, and vehicle routing among independent dispatch centers. These automated negotiations can be conducted from a cellular phone through mobile agents. The cellular phone just invokes a mobile agent, disconnects from the network, and the mobile agent hops around the network (wired or wireless) negotiating on your behalf.
- Collaborative agents that can serve as the mediators in manufacturing supply chains. These agents can monitor the status of supply chains, detect delays, and find alternative sites in case of failure/unacceptable delay of a supplier. Supply chain systems of this type are proactive in nature and are known as "Zero Latency Supply Chains" because they can detect and correct problems without any delays (hence zero latency).
- Multi-agent systems for large scale trading and brokering that involve many local agents (some static, some mobile). Local agent managers handle local agents and multi-agent systems handle multiple local agent managers. Examples of multiagent systems can be found in the energy market where multiple energy suppliers can have their own local agents coordinated by multi-agent systems and the manufacturing segment for manufacturing resource planning. For discussion and examples of multi-agent systems, see the Communications of ACM special issue on this topic, May 1999.

12.6.3 Generic Architecture of Mobile Agent Environments

Before discussing security issues, it is a good idea to look at the architectures of mobile agent environments. Although the environments vary widely, most current mobile agent environments are based on Java due to its portability and mobility features. Most mobile agent environments use an architecture that is somewhat generic. Figure 12-21 shows a generic mobile agent architecture discussed by [Wong 1999] that can be used as a framework for discussion. This architecture shows six different components:

- The agent manager is responsible for sending and receiving agents to/from remote hosts. It serializes the agent and its state before sending it to remote hosts and also deserializes and receives the agents on the other end.
- The reliability manager makes sure that the sent agent is properly received by the remote host. It also guarantees the persistence of state associated with agents.
- The security manager authenticates the agent before it is allowed to execute at the receiving host. All other mobile agent system components interact with the security manager to authenticate and authorize mobile agents.
- The application gateway provides secure interactions with external applications (non-agent) such as purchasing systems and product catalogs.
- The directory manager keeps a directory of all agents in the network.
- The interagent communications manager is responsible for managing communications between multiple agents that are dispersed throughout a network.

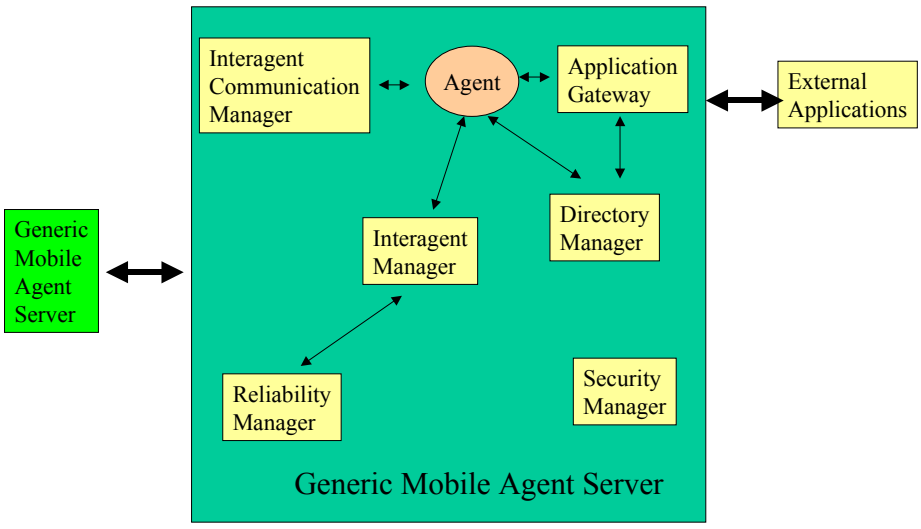


Figure 12-21; Generic Mobile Agent Architecture

An important implication of this architecture is that a mobile agent server must exist at each site where the mobile agent is supposed to run, Thus if you have a network with 100 hosts but only 7 have the mobile agent server, then your mobile agent can only roam around the 7 hosts. In addition, mobile agent systems are not interoperable (so what is new!). Thus if you are using the Aglet mobile agent system, then the aglet mobile agents can only run at the sites where the Aglet server is running. Considerable work in multiagent systems and agent transfer protocols (ATPs) is needed to address these problems (see the Communications of ACM special issue on multiagent systems, March 1999 and ATP specification at the IBM site http://www.trl.ibm.co.jp/).

12.6.4 Security Issues in Mobile Agents

We can discuss mobile agent security by using client and server side security analysis.

Mobile Server Security. A major concern specific to mobile agents is the protection of the servers running the agents. Running arbitrary programs on a machine is dangerous: a hostile program could destroy the hard drive, steal data, or do all sorts of other undesirable things. This risk must be thoroughly addressed if mobile agent environments are to succeed. Two types of security approaches are possible to protect servers from malfunctioning and hostile agents: physical and social.

- Physical security refers to building servers for agents in such a way that the agents cannot harm the server. The ``laws of physics" of the server execution environment can be designed to make dangerous operations difficult or impossible. Common approaches involve creating a ``sandbox" for visiting agents, restricting access to resources (preventing disk writes, for instance) and ensuring the agent cannot escape those restrictions. This approach to security is attractive; when it works, it is entirely effective. But the viability of physical security in the face of design complexity and server implementation bugs is unclear. In addition, physical security is typically focused on protecting some underlying aspect of the server from the sandbox the agent is trapped in. But if multiple agents are put in the same sandbox how can the server guarantee that one agent cannot harm another? As we put more trust in the computations that take place inside sandboxes, the security of those sandboxes themselves becomes important.
- A second approach to server security is using social enforcement mechanisms to punish the creators of harmful agents. If a server administrator can find out who is responsible for a malicious agent, then that person can be held accountable via social mechanisms (such as lawsuits). Digital signature technology makes identifying the authors of agents possible. But there are limitations to a purely social approach to security. It may not be clear which agent is responsible for damage, nor will it be easy to determine ahead of time which agent authors are trustable. In practice some combination of social and physical enforcement of server security will be useful.

Agent Security. The complement of server security is agent security: whether the agent can trust the server on which it is executing. A mobile agent might contain secret information such as proprietary data and algorithms. Worse, servers might have an incentive to subvert the computation of a visiting agent. In the Internet-based DES cracking effort currently under design a major concern is protecting the computation from sites that pretend to do pieces of the problem but return false answers [Tre96]. Physical security answers to this problem are difficult. Secure, trusted hardware on the server could guarantee agent safety but is unlikely to be widely deployed. Agent programmers can protect their agents by obfuscating their code and verifying the results of the remotely-performed computation but the general applicability of these techniques are unknown. Social solutions may be possible in the form of reputation systems for servers. This area of security has largely been unexamined.

12.6.5 Mobile Agent Security Summary

While mobile agents are a useful approach to distributed computation, in practice they have not been used in many real commercial applications (most mobile agent applications are research prototypes). There are many technical challenges to implementing large-scale industrial-strength mobile agent systems. Most problems are centered around the issues of security and reliability. Basically, mobile servers must be designed, implemented, and deployed that not only allow mobile agents to run, but allow

them to run safely. Another key area of work is multiagent systems that involve coordination between multiple mobile agent systems. Mobile commerce presents an interesting area of work for mobile agents and may provide the "killer application" that mobile agent systems need so badly. To succeed, however, security weaknesses must be addressed.

12.7 Email Security – S/MIME and PGP

Most e-mail client and server programs use Internet systems such as SMTP send e-mail as clear text. The **Secure Multipurpose Internet Mail Extensions (S/MIME)**, a specification for secure electronic messaging, can be used to prevent the interception and or forgery of e-mails. S/MIME is an extension of MIME and is based on technology from RSA Data Security. S/MIME supports encryption of email messages for privacy and digital signatures for authentication. S/MIME incorporates three public key algorithms (DSS, RSA, and Diffie-Hollman). For message integrity, it uses MD5, among others, and triple DES is used for encryption. Information about developments in S/MIME can be found at http://www.ietf.org/html.charters/smime-charter.html.

PGP (Pretty Good Privacy) is a popular program that is used frequently for email security. PGP uses symmetric as well as asymmetric encryption (it uses a 128 bit key to encrypt files or messages). To send secure email by using PGP, the sender encrypts the message by using a private session key K and sends to the receiver. The session key K is sent by the sender also after it has been encrypted by using the receiver's public key so that only the receiver can decrypt the key. We briefly reviewed PGP in the previous chapter. Information about PGP can be found at the web site: www.pgpi.com

PGP and S/MIME are very similar in many respects and use the same type of cryptographic techniques. They are both on the same IETF standards track also. However, it appears that S/MIME will emerge as the standard for secure corporate email while PGP will stay as the favorite choice for secure personal email.

In addition to securing email for privacy purposes, it is important to protect enterprise email system against email viruses and attacks. The most deadly viruses are typically distributed worldwide via email in a matter of hours (for example, the LoveLetter virus). Email worms and viruses can reach enterprise systems and infect corporate users through harmful attachments. Some viruses are transmitted through harmless-looking email messages and can run automatically without the need for user intervention (like the Nimda virus). We will discuss viruses and other forms of malicious code in section 12.9.

12.8 Additional Application Security Issues

So far we have examined the e-commerce/e-business, mobile application, and email security. For completeness, let us now quickly review the security approaches for the following applications:

- Applications based on CORBA can employ the CORBA security software discussed in the previous chapter. CORBA security is built on top of SSL.
- Applications that employ application servers such as commerce or B2B servers should very carefully evaluate the security features of the application servers before building several applications on top of an application server. We discussed this topic in the previous chapter.
- Applications based on the Microsoft .NET platform should use the .Net security features. In particular, the concept of Passport in .NET provides a single sign-on and a certificate authority. This feature should be carefully evaluated. We discussed this topic also in the previous chapter.
- Applications based on the Sun J2EE platform can use the J2EE security features that include SSL and also a combination of other security packages .

12.9 Malicious Programs and Viruses

12.9.1 Overview

Malicious programs exploit vulnerabilities in computing systems to perform undesirable operations. Some of these malicious programs are self-contained software routines that can be scheduled and run independently while others are code fragments that are attached to another utility or programs such as email systems. The problem is that these programs, usually in the form of viruses, are on the rise. According to an Information Week Research report in July 2000, the bill to 50,000 US firms in 2000 for viruses and computer hacking amounted to $266 billion. By comparison, in 1999, computer viruses cost US business $12 billion – an increase by a factor of 20. Typical malicious programs can be classified in terms of the following:

- **Virus.** This is a program that "infects" other programs by modifying them. These modifications infect other programs in a similar manner, thus making a family of programs "sick". This is one of the best known examples of malicious programs.
- **Logic Bomb**. This is one of the oldest malicious programs. It basically is malicious code that sits inside a program silently and then "explodes" when certain conditions occur. Examples of conditions can be a time and date, opening a certain file, or a certain command.
- **Trap Door**. This type of malicious code takes advantage of some hidden ways of getting into the system. Trap doors are legitimately used by developers and testers to debug programs and systems but can be deadly in the hands of intruders.
- **Trojan Horse**. This is an apparently useful program or utility that has some hidden agenda such as open a trap door or to disable a monitor to help potential intruders break into a system. For example, a utility that does backup/recovery on a system could be replaced with a Trojan horse that recovers (replaces) a system administrator file to make it easier for intruders to break in. Trojans are typically emailed to the victim, often disguised as an attractive attachment, such as a joke, to convince the recipient to run the program.

12.9.2 Common Means of Introducing Malicious Code

E-mail is by far the most common distribution mechanism for introducing malicious code. Well known examples are the email viruses transported through Word macros (e.g., Melissa), infected attachments (e.g., Love Bug) and commands embedded in HTML mail. In addition, email is used specifically to launch spies to obtain confidential information in organizations.

Many of these viruses have cute names - like the Visual Basic (VB)-script worm Love Bug, the macro virus Melissa, and the Explore Worm. Email is a prime means for installing backdoors (Trojans) and other harmful programs to help potential intruders break into a corporate network.

There are numerous examples of email attacks targeted to damage or disrupt large companies such as Microsoft. For example, an email attack on Microsoft's network was launched in October 2000 through a backdoor Trojan virus maliciously emailed to a network user. This is, unfortunately, a global phenomenon. Earlier in 2000, Japanese police arrested a hacker who was emailing viruses to a large company to cause its computer system to shut down.

HTML viruses, also known as **active content attacks** or browser attacks, do not require user intervention. These attacks tend to use the scripting features of HTML or of the email client to execute malicious code on the recipient's computer. These attacks can be used to display undesirable content or lock the machine for Denial-of-Service. Unlike the email-based viruses such as Melissa that required the victim to execute them by opening the infected email attachment, the HTML viruses do not require user intervention in order to be activated.

Buffer overflows are also used to execute malicious code. For example, an array with sizes larger than the program allows can be developed in the program to cause overflows. However, the overflow area has some code that can be activated from other programs. Sophisticated developers with knowledge of machine code can launch these type of attacks.

Future attacks could be launched by taking advantage of XML, RDF and DOM weaknesses, as discussed in previous chapters. In addition, the layers of middleware services, also discussed in previous chapters, can provide haven for clever hackers (let us hope not!).

Development of viruses at present is not difficult. Sites such as www.SecurityFocus.com provide adequate information.

12.9.3 What Can Be Done?

Successful approaches to deter malicious attacks consist of many software packages plus administrative procedures because no one approach is sufficient to handle the wide range of attacks. Examples of the approaches are:

- **Harden the system software**. Hardening the operating system and the servers before production use, discussed previously, should take care of many trap doors and other vulnerabilities.

- **Install firewall(s)**. Network firewalls can prevent access to corporate networks by unauthorized users. A host firewall, described earlier, may also be needed. The main limitation of the firewalls is that they do not check the content of mail being sent and received by those authorized to use the system. This means that email viruses can still pass through the firewalls.
- **Install virus checking software** and update it frequently. This is a good approach but anti-virus software cannot protect against new viruses and attacks because the vendors cannot quickly update their software to deal with new viruses. For example, the LoveLetter virus and its variants were distributed worldwide via email in a matter of hours and caused major damage before the anti-virus companies could respond to them.
- **Install special virus content checking software** for all inbound and outbound email at email server level, before distribution to users. Thus potentially harmful content can be detected and removed from a dubious email. GFI MailSecurity for Exchange/SMTP is an example. Another example is the email virus checking and testing tool available at http://www.windowsecurity.com/emailsecuritytest/. This testing tool, developed specifically for MS Windows environments, conducts ActiveX vulnerability test and variety of tests on various aspects of MS Outlook Express and MIME.

In addition to the software, the following practices should be adopted:

- *Never* open any files or macros attached to an email from an unknown, suspicious or untrustworthy source. Delete and purge these attachments immediately.
- Delete spam, chain, and other junk email without forwarding to other people in your enterprise.
- Do not download files from unknown or suspicious sources.
- Always scan a floppy diskette from an unknown source for viruses before using it.
- Back-up critical data and system configurations on a regular basis and store the data in a safe place.
- Avoid direct disk sharing with read/write access with others unless there is a strong business need to do so.

12.10 Short Case Studies and Examples

12.10.1 Centex Avoids the Nimda Worm

Centex Title and Insurance is part of Centex Corporation's Financial Services group. With about 800 employees located in 72 offices, and annual revenues of roughly $80 million, Centex Title and Insurance has many customers who rely on its Web site for critical information they often use to make major life decisions. The company provides title insurance, escrow services, commercial and personal insurance, and real estate information services. To trace the origin of titles for homes and other property the customers want to buy, Centex does extensive research and maintains/transfers very sensitive information in the process. There is little margin for error when providing services that can impact where and how the customers live. In addition, Centex took pride in the fact that Salomon Smith Barney designated Centex as one of the Top 10

Internet-savvy companies in 2000 and that Fortune magazine rated Centex as "America's Most Admired Company" in the engineering and construction category, three years in a row.

Centex has many customers who need to access its Web site to find highly time-sensitive data. A security breach that affected Centex Web site availability, performance, or content would be particularly embarrassing for a firm like Centex. In October, 2001 suspicious activity was observed on the company's Web site (www.centextitle.com). When the company management heard that this was a Nimda worm attack, they were very nervous. Coincidently, about a week before the attack, the company had installed a G-Server from Gilian Technologies on the network for added security. The system is based on a technology called ExitControl, in which data is inspected as it leaves the Web server. The program uses digital signatures to verify the validity of this data; any defaced or otherwise modified pages are not allowed to pass through the system. Instead, the device sends an encrypted, archived copy of the original page. The question was: is the G-Server going to help or hurt the situation. Fortunately, the G-Server worked as planned, and none of the corrupted pages leaked out of Centex's site. The security device from Gilian Technologies (www.gilian.com) had prevented the worm from wreaking havoc on the Web site. The technical team was able to rid the site of the problem within two or three hours.

Here are a few more details about the G-Server (see www.gilian.com for additional information). The G-Server is a rack-mountable device that sits between the Web server and the external network, and uses digital signatures to authenticate the outgoing content. It monitors all outgoing content from the Web server and compares it to the digital signature the author created for that content during publication. When the signature is created, the system also generates an archive of each object, including static and dynamic HTML pages, CGI, GIFs, JPEGs, and so on. The G-Server automatically uploads new pages and other objects to the server, and automatically generates a digital signature for each object. If the signatures of the original and newly requested page match, the page is released. If the system detects a discrepancy in the signatures, however, the G-Server replaces the questionable page with the archived copy of the original, then alerts the administrator of the problem via e-mail, phone, or pager. When content is revised or updated, it's reauthenticated to help ensure that no unauthorized changes have been made. All communication with the G-Server is Secure Sockets Layer (SSL)-encrypted. The G-Server integrates with software from BMC, Tivoli, and Check Point. According to Gilian, the G-Server introduces 2ms of latency and has a throughput limit of 80Mbits/sec in the worst case. The system supports IIS, Apache, Netscape, and iPlanet Web servers.

Sources:

- Elizabeth Clark, “Business Case: Centex Nixes Nimda at the Exits”, Network Magazine, Sept 5, 2002.
- www.centex-title.com
- www.gilian.com

12.10.2 e-Business Security at an International Financial Institution

As part of its e-business initiative, a U.S. headquartered international financial institution wanted to find a process to allow its commercial customers to access banking applications. Although passwords and ID's were being used, they were not viewed as a long-term security solution for authentication in a B2B, B2C, and B2E (business to employee) settings. An IT Director thought that PKI could potentially facilitate business and build trust with retail customers. A pilot program was launched to work with retail online banking applications and build an authentication process for an online banking system. It was well understood that PKI is not just about technology, but about the rules around the technology. The implementation partner, one of the Big 5 Accounting Firm, assisted the client in writing the certification practice statements (CPS) and certificate policies (CP). A proof of concept PKI pilot project was developed for the retail online banking system.

Although the project was successful, it was later determined that it was not feasible to rollout PKI for the retail sector because of financial and organizational constraints. However, it was determined that the completed work was leverageable for the commercial sector. Consequently PKI is now being used primarily in business to business (B2B) processes ranging from currency trading applications, to domestic check clearing applications. It has been decided to implement PKI first as an external project that concentrates on customer-facing applications. This external project is targeted to migrate to an internal process, including email, for its over 70,000 worldwide employees. For implementation, a vendor was chosen to scale with the client as business processes, and consequently PKI needs, expanded due to more commercial banking applications and potential mergers.

Source: "The Evolution of e-Business Security Requirements" Gartner White Paper Prepared for Verisign, Inc., Engagement: 220011370

12.10.3 Yellow Corporation Detects Intrusions in its eBusiness Operations

Yellow Corporation, a Kansas-based Fortune 500, is one of the largest freight carriers in the nation. In the freight business, timely availability and integrity of information are crucial for the customers as well as carriers. Due to this, Yellow Corporation made a strategic decision during the mid-1990s to migrate many of its routine business transactions to the Internet. Over time, Yellow has embraced e-business and is using the Internet to broker many of its B2B transportation and related services. Specifically, the company has made a concerted effort to move most of its day-to-day customer and driver interaction to the Web. Yellow's online efforts were recognized by CIO magazine, which named Yellow as one of the Top 100 technology companies in 1999.

Yellow customers use the Internet to get instant rate quotes, which include all freight charges, fuel charges, and projected delivery dates. The customers can schedule pickups and delivery, trace shipments, and make payments electronically. Corporate and independent carriers can get needed information regarding available shipments. print

necessary forms and permits, and gain immediate access to Yellow's customer service department by using YellowLive -- an AOL Instant Messenger-based service.

Security has been a major concern for Yellow for its e-business operations. To secure its operations, Yellow selected eTrust Access Control. This tool secures sensitive system files and key critical data files, serves as a firewall, and also control and manage their UNIX systems from a single point. This tool allows customers and carriers to have free and easy access to the information they need while protecting the more confidential operations of Yellow's enterprise servers.

The investment in eTrust Access Control paid off. During the summer of 1999, a Yellow Information Security Analyst noticed a series of alerts originating from the company's UNIX Web servers. He checked the log files on the servers for the cause of the alerts, and found that an unidentified user had tried to gain access to Yellow's proprietary system files and user IDs via the Internet. The alerts showed that someone, who doesn't have enough system knowledge to operate at a system level, was running a scanning type of program like 'nmap' -- one of many hacker programs. It was estimated that the attack lasted for 60-90 minutes, during which time the intruder scanned various server ports, looking for access to any of the services that were available on the server, including ftp, telnet, http, and e-mail.

The intruder used a blunt force hack by running the script, trying to exploit the services that were on that server. The firewall component of eTrust Access Control locked up the basic services on the Yellow UNIX server so the intruders were denied access to everything they tried. Later, checking the IP address of the intruder, the analyst found that it had been an individual from outside the company. Further investigation with the ISP determined that the suspect account had since been closed.

The attack on Yellow's corporate servers, if successful, would have experienced a loss of availability that would have transferred into financial loss. It was estimated that the company could easily have lost about US$30,000-US$60,000 an hour.

Source: http://www.ebusiness-security.com/Case%20Studies/Access_control_case_study.htm

12.11 Suggested Review Questions

1) What are the key issues in application security and why are they important?

2) Suppose you need to secure a Web-based inventory management system. Explain the tradeoffs between network, platform, and application security for this application.

3) What are the main issues in client side security?

4) What is SET and how is it used in e-commerce? How is SET related to SSL?

5) What are the main security issues in mobile applications and what type of solution technologies are available to address these issues?

6) Compare and contrast PGP with S/MIME for email security.

7) What are the categories of malicious code and what approaches will you use to protect against such attacks?

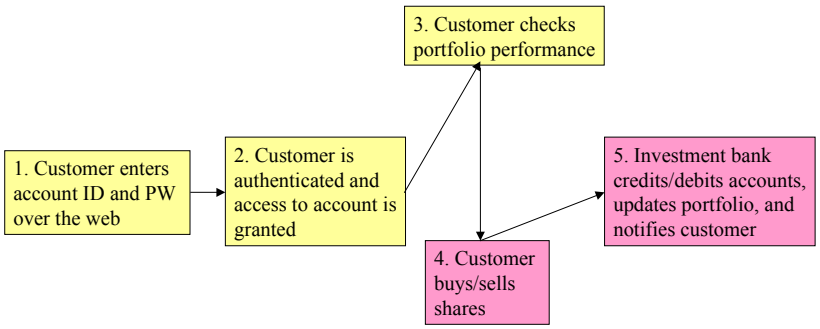
8) Complete the following table:

Application	Unique Security Risks	Solution Approaches
Online purchasing	▪ ▪ ▪ ▪ ▪	▪ ▪ ▪ ▪ ▪
CRM	▪ ▪ ▪ ▪ ▪	▪ ▪ ▪ ▪ ▪
SCM	▪ ▪ ▪ ▪ ▪	▪ ▪ ▪ ▪ ▪
B2B eMarkets	▪ ▪ ▪ ▪ ▪	▪ ▪ ▪ ▪ ▪
Mobile Commerce	▪ ▪ ▪ ▪ ▪	▪ ▪ ▪ ▪ ▪
Positional Commerce	▪ ▪ ▪ ▪ ▪	▪ ▪ ▪ ▪ ▪

12.12 PART IV Case Study Revisited: NRW Adopts Web Services and Mobile Applications

12.12.1 Overview

NRW has ventured into Web Services and mobile commerce. Due to business pressures, NRW is also thinking of extending Account Balance Program to go beyond just checking the balance. NRW wants its customers to also buy and sell stocks online and thus make it a complete e-commerce system with payments and adjustment being made by the bank (see Figure 12-22). This added feature, NRW feels, will significantly increase customer loyalty but may introduce some security risks.



Note: Steps 4 and 5 are new

Figure 12-22: Extending Account Balance Program

In addition to Web Services and e-commerce, NRW is seriously considering wireless access to the accounts. Despite several concerns, they want to investigate the access to the account from mobile users within the organization and also access from cellular phones.

12.12.2 Risk Analysis

Due to the technology stacks involved in supporting these new services, we need to first look at architectures to understand the risks. Figure 12-23 shows an application architecture for the NRW system that captures the application and middleware choices (the networks will be introduced later). It is assumed that the new functions of buying/selling and investment bank handling is performed by the back-end applications.

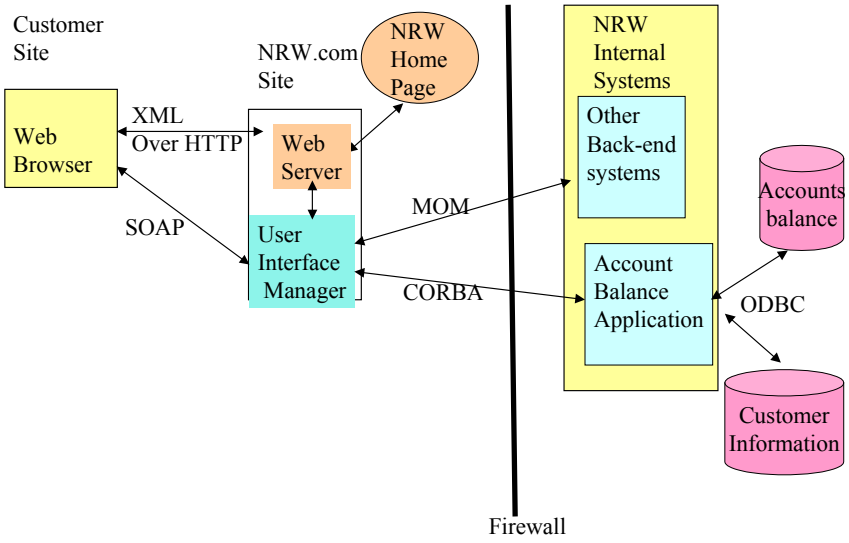


Figure 12-23: Web Services Architecture of the NRW System

This architecture uses Web Services and shows how the customer accesses the system over HTTP and then uses SOAP to invoke a protected user interface manager (UIM). A firewall protects the internal corporate resources. The UIM accesses the NRW internal systems through middleware services (CORBA and MOM) over this firewall. This view can be further mapped into .NET or J2EE views, if needed. For example, for J2EE, the UIM can be an EJB (Enterprise Java Bean) that is invoked through a servlet and for .NET, the UIM is a .NET managed component that can be invoked through ASP.NET. NRW is leaning towards .NET. Naturally, WSDL and UDDI may be used to advertise and locate the UIM before it is invoked through SOAP.

Figure 12-24 shows a more detailed physical architectural view of the NRW system that shows the application, middleware, local software, as well as network components. This view is a refinement of the view presented in the NRW case study at the end of Chapter 9 (Part III). This view shows that the new e-commerce programs for stock buying/selling and investment bank handling are allocated to the mainframe, and the ABP is allocated to the windows NT with a Web server. In addition the middleware components (MOM, CORBA, and XML-SOAP) are shown. The wired and wireless networks with firewalls are also shown.

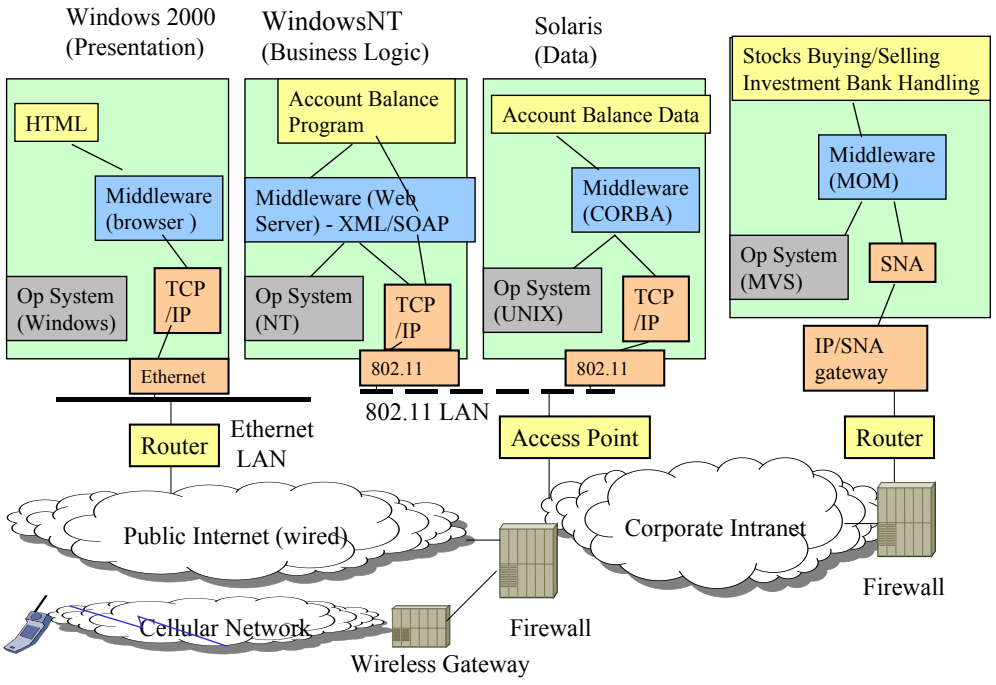


Figure 12-24: Detailed Physical Architecture of the New NRW System

Securing this complex jungle is a daunting task, but we have been taking a step by step approach. We already considered the wired/wireless network issues at the end of Chapter 9, so now let us now review the platform and application issues.

For platforms, the following risks need to be considered:

- The weaknesses in the operating systems (Windows NT, Solaris, and MVS) need to be addressed.
- The Web Server problems and the weaknesses of Web Services-XML need to be understood.
- The security of various middleware services (e.g., CORBA, MOM) need to be understood.

For applications and databases, the new stock buying/selling and investment bank applications need to be highly protected because compromise of these applications could have a disastrous impact. In addition, the "older" accounts balance program and database need to be protected.

12.12.3 Risk Mitigations and Circumventions

NRW will establish some key attributes of the security architecture for the online investment program. In addition to UIM, the ABP (Accounts Balance Program) will be heavily protected. The main elements of the countermeasures are use of encryption, Virtual Private Networks (VPNs), firewalls, authorization, and general attempts to minimize the points of access to critical databases and applications. Security Assertion Markup Language (SAML) will play a key role. All external users, customers, remote employees, global partners and suppliers, have to receive certificates in order to access

the corporate information. Specifically, the countermeasures should include the following:

- XML Encryption and SAML should play a key role in this project. Because the SAML standard is designed only for the exchange of secure sign-on information between parties (users, machines), it can be used to allow issuing parties to use their own chosen methods of authentication (e.g., PKI, hash, or password).
- There should be single user name and logon. Once a person is logged on to the NRW system, they will not have to log on again. This can be facilitated through SAML, or .NET Passport.
- The users' authorization to the network will be stored in a web server/directory server. The NRW users' rights to what files and directories they can access, will all be located and accessed from this server. This is sufficient for static web content security. The application will provide further authentication and authorization once the network access has been granted to the NRW employee.
- Proper UDDI and WSDL security will be needed if the ABP is advertised as a Web service.
- The platform for NRW's Extranet system has to be highly controlled. This means that only NRW's data center personnel has physical access to NRW's application server and network equipment. If a business partner owns a piece of equipment it is to be shared between both organizations.
- Firewalls will be used to localize the different areas of the firm's security architecture. The firewall will separate the corporate web site from the ABP, the customer database as well as the investment databases.

For e-commerce, NRW needs to make sure that the attackers do not use a Web browser over the Internet to invoke weak server side programs (CGI scripts, servlets). In fact, it is important for NRW to make sure that the system weaknesses cannot be exploited by the attackers to gain privileged status. In case of e-commerce, privileged status users can possibly buy and sell shares. The best way to defend against such attacks, as stated previously, is not to leave untested and flawed code on the machines.

For wireless and mobile access, NRW will be potentially exposing itself to several security risks such as the following:

- Denial of service can happen due to any number of reasons. For example, wireless network outages and network flooding, viruses, hackers and physical equipment problems could all deny users the ability to conduct business with NRW.
- Wireless access from internal and external users will expose the company to several additional assaults of privacy, integrity, authentication, and others.
- Unauthorized wireless users may access information on customers, accounts and research information.
- Attacks using falsified authentication and Man-in-the-Middle can be launched over wireless networks.

To address these issues, NRW needs to take the following measures:

- Very strong encryption, authentication, and authorization techniques should be used at the application level for wireless users. Digital signatures and digital certificates should be used for authentication.
- The internal wireless LAN should be placed outside the firewall. Thus any internal access from a wireless LAN will have to go through the firewall.

- Virtual Private Network (VPN) should be used on top of the wireless network and SSL should be employed between wireless users and NRW resources.
- WTLS (Wireless Transport Layer security) certificates should be used at handheld devices and migration to 3G cellular should be expedited to take advantage of the 3G security.
- Rigorously defend against e-commerce attacks by removing all untested and flawed server side code and stringent firewall policies.