SETLabs Briefings
ENTERPRISE ARCHITECTURE & BUSINESS COMPETITIVENESS VOL 2 NO 4
Oct – Dec 2004

Threat Modeling in Enterprise Architecture Integration

By Visveswaran Chidambaram

As integrated systems are becoming more complex, vulnerability analysis is crucial to assess and safeguard against threats

Enterprise Architecture Integration (EAI) has matured over the years to enable limitless information sharing across the globe and across a multitude of platforms. It has, in the meanwhile, resulted in heightened security concerns among enterprises. Given the almost hostile environment in which EAI technologies function today, it is not surprising that threat modeling is fast becoming a key milestone in the requirement and design phases. Unlike other aspects of requirement gathering and design, threat analysis is not a one-time job. Continuous feedback to the threat modeling mechanism from the systems of their interaction with the outside environment, latest technology updates, and process reengineering are crucial for the successful implementation of a threat modeling system.

The approach to building a threat modeling system will involve the following steps:

1. Creating an architecture overview.
2. Creating a security profile of the system by breaking it down to its functionality and network topology.
3. Creating an Asset Classification List (ACL).
4. Performing vulnerability analysis and identifying threats.
5. Rating the threats.
6. Evaluating counter measures.

Note: This article only provides pointers to readers on approaches to threat modeling and does not attempt a detailed analysis of the same.

ARCHITECTURE OVERVIEW AND TOPOLOGY DECOMPOSITION

Consider the example of a standard implementation of an EAI scenario (Figure 1) in a company that provides shipping services. The company's website accepts orders from the

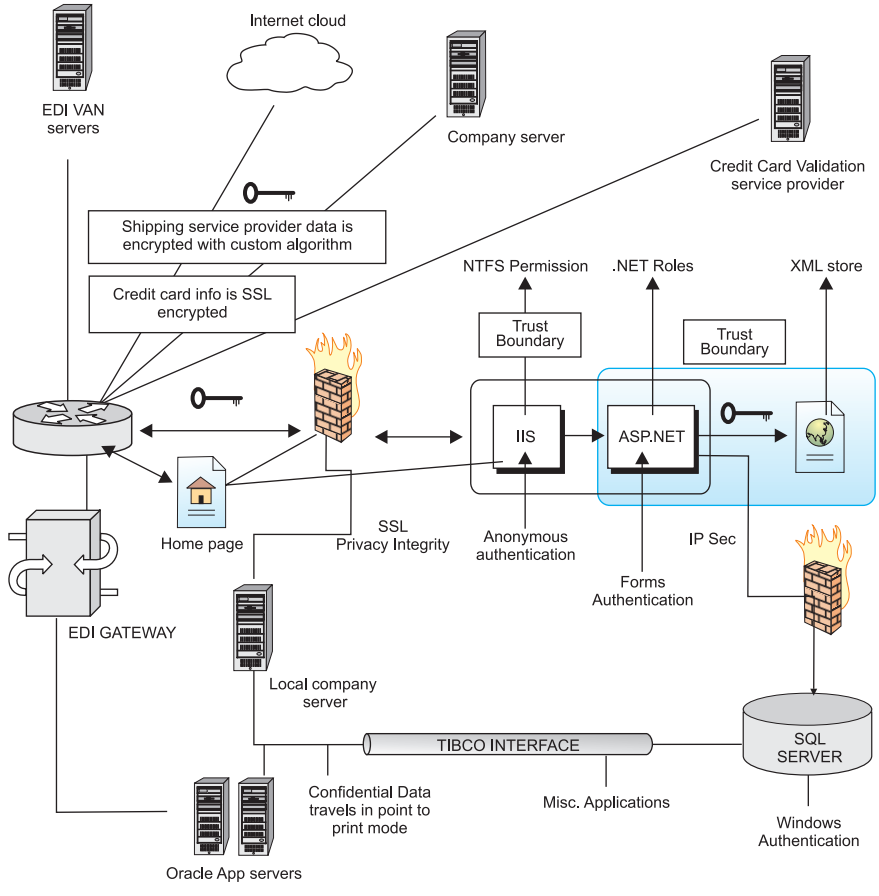

Figure 1: *A sample EAI scenario* ***Source:*** *Infosys Research*

outside world. It also accepts orders through an electronic data interchange (EDI) interface that gets directly pipelined into the Oracle application servers. The web orders are processed by web servers that are placed in the demilitarized zones (DMZs) , which are written on to a temporary XML store first, and then on an SQL database inside the company's network. The processed orders are then transmitted to the Oracle backend to be fulfilled. This is done through a TIBCO middle layer. The fulfillment details are transmitted back to the SQL server to be displayed on the website. Orders that need to be shipped are assigned a tracking number that is generated by the shipping service provider's server, which is situated inside the company. This server then transmits the details to its site so that the package can now be tracked. The orders received through the EDI interface are similarly treated and processed by Oracle and fulfillment details are sent back to the customers through the EDI gateway. The sample system has been intentionally kept simple to facilitate the understanding of security analysis.

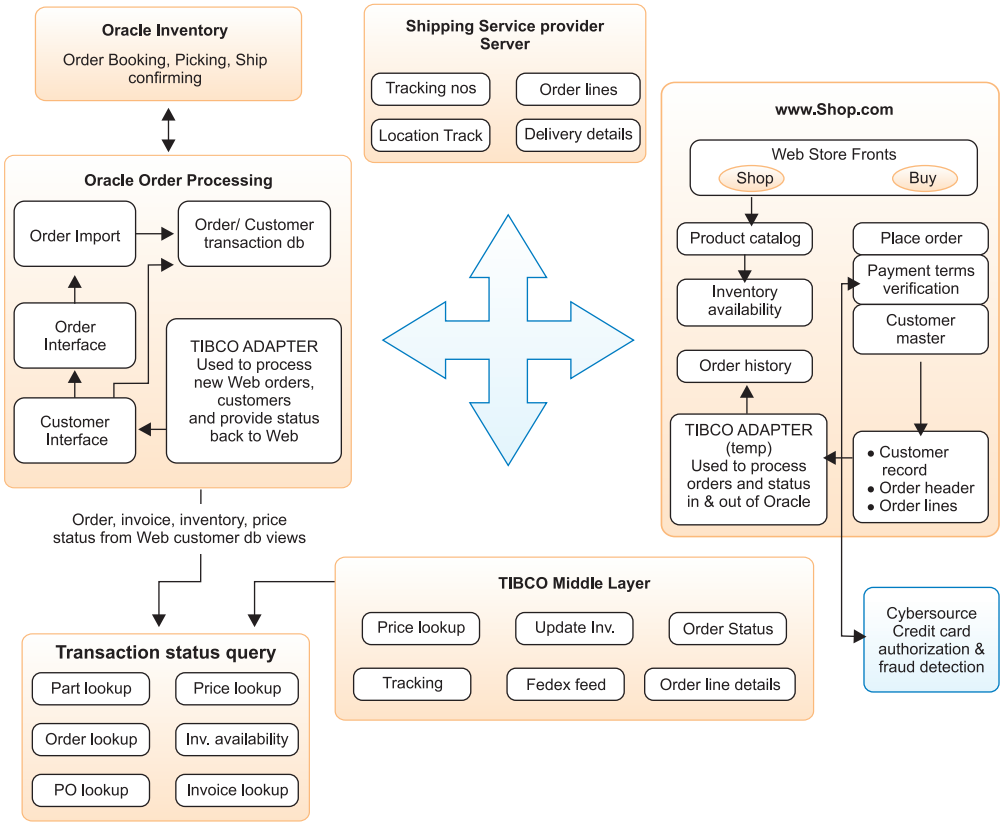

Figure 2: *It is important to map the data flow to determine trust boundaries and for asset classification.* ***Source:*** *Infosys Research*

A different perspective of the same architecture (Figure 2) attempts to map the data flow against the physical servers and network topology. The key purpose of this exercise is to determine trust boundaries that surround each of the assets in the architecture and to form an input to the next step in the process, that is, asset classification. In order to map trust boundaries it is necessary to know whether any step in the process can depend on the input of its upstream process or user input as the case may be. For example, an input from an external server that authenticates itself by means of a digital certificate is more trustworthy than input from a user that comes in as free text on port 80.

In the sample architecture, user data is accepted through a website; processed by the web server running IIS and .NET remoting classes, which validate the data before storing the orders in the SQL database.

The web server is inside the DMZ while the SQL server is situated on the company's network. The order data received by the SQL server is sent to the Oracle order entry system by a TIBCO interface. The company also accepts orders through an EDI interface. The data is

piped into an EDI gateway that is converted and validated by a customized interfacing program, and then sent into the Oracle order entry system.

ASSET CLASSIFICATION

Identifying EAI assets is crucial because it forms the foundation for the threat modeling exercise of the enterprise. An asset inventory needs to be drawn up and assets classified based on their relative importance to the business as well as information.

A web server for example, would form an important asset because it is impossible to keep a website running without one. A database with credit card information would be another important asset in terms of confidentiality of information. Both the above assets would be listed high in an asset classification exercise. The main aim of this exercise is to determine how resources can be efficiently apportioned into protecting these assets.

Assets include information assets such as customer data, network assets such as routers and firewalls, infrastructure assets such as servers, gateways, and application assets such as the code that provides business functionality, and interfacing programs that transfer data between two disparate platforms.

After they are identified, the assets must be classified by taking into consideration three important parameters: Confidentiality, Integrity, and Availability. Asset classification involves rating the assets on the basis of the above three parameters. Consider the example of the web server in the DMZ, which receives data from the customer. It is the first point of interaction for the customer with the order processing system. It therefore needs to be available 24x7 to be able to accept orders. The web server is highly exposed to the internet making it very vulnerable to a denial-of-service attack. Thus it would be rated very high with regard to availability. Next, the customer having entered an order would enter credit card details. This means that this server would also need to ensure the confidentiality of the data that it is receiving. Also, it is possible for a hacker to not only intercept this data, but to modify it. To rule out the possibility of the hacker changing the delivery address and order lines to suit his/her needs, the integrity of the order needs to be safeguarded as well. Keeping the above analysis in mind, the web server would receive the highest rating in an asset classification exercise.

Not all vulnerabilities are threats; only a weakness that can be exploited to harm the network is a threat

Similarly, all the assets in the inventory need to be ranked in terms of their rating. This ensures that the available resources are appropriately segregated to facilitate asset protection.

A cost-benefit analysis can be done based on the service-level commitments and resources spent on protecting the assets to deliver such a service. Appropriate changes to the architecture can be accordingly made in due course. For

example the decision to host an SQL database in the DMZ as opposed to within the company's network is one decision that would purely depend on such a cost-benefit analysis.

Classifying the web server that is in the DMZ as 'at high risk' may be an easy choice to make. But consider a road warrior who uses a laptop to accept orders from customers and then connects to the company's internal network to upload data to the database. Since his laptop is a mini version of the website functionality, it is reasonable to assume that he would have an SQL server loaded on his machine. Since he also carries his laptop with him it is also possible he has connected it to the internet using a local service provider. If the virus signatures on his machine are not updated or if the latest patches have not been applied to his machine then the laptop becomes vulnerable to viruses that are active in the wild. Now, under these circumstances, it is very probable that he can infect the internal network of the company while he is transferring orders into the master database. An attack need not always come from the outside; employees can perpetrate it from within. This alone would result in a business development manager's laptop being rated very high in the asset classification list, though it actually connects to the network from the inside.

Another very important asset that needs a close look is the server that hosts the public key infrastructure (PKI) keys. If this server is compromised then all the encrypted data that is being sent on the internal as well as external networks are at risk. This server has a higher probability of being hacked into by an employee than an outsider and hence must be rated very high in the asset classification list.

IDENTIFY THE THREATS

The next step to asset classification and ranking is vulnerability analysis and threat identification. The difference between vulnerability analysis and threat identification is that, while the former examines all the weaknesses that the asset is susceptible to, the latter actually maps these weaknesses to the environment in which this asset lives. In other words, it involves the security context of the asset and involves analyzing how many of these weaknesses can actually be exploited. Not all vulnerabilities are threats; only a weakness that can be potentially exploited to harm the network is a threat.

The factors that determine a threat are the security context of the asset, the probability of the vulnerability being exploited, and the risk appetite of the organization. (Risk appetite is an entity's attitude towards risk. A high-risk appetite indicates an entity's willingness to live with a higher level of risk.)

The web server in the sample application has already been ranked high in terms of confidentiality, integrity and availability. Being exposed to the internet, handling sensitive data both in terms of the order for the company and the credit card details for the customer, makes it vulnerable to attacks. The probability of this vulnerability being exploited is very high. A few of the high-impact threats to this web server are denial-of-service attacks, session hijacking, and break-ins. Correspondingly, a higher portion of the available resources must be spent in protecting these servers.

A good approach toward threat identification is Microsoft's STRIDE. (This is a goal-based approach, where an attempt is made to get inside the mind of the attacker by rating the threats against **S**poofing identity, **T**ampering with data, **R**epudiation, **I**nformation disclosure, **D**enial of service and **E**levation of privilege.)

A different approach is to use an exhaustive threat list and walk through them one by one for

each and every asset and evaluate them in the light of the assets' ranking and the security context in which the assets operate.

TYPES OF THREATS

Threats can be classified as follows:

- Network threats
- Server or Host threats
- Application threats

Network threats: In the sample system of the shipping service provider, some of the major network threats are:

- Web servers being subjected to a denial-of-service attack.
- IP spoofing, where it is possible to impersonate a customer and steal or modify information.
- Faulty configuration of firewall rules allows outsiders to get access to the database and change the data.
- Errors in access control lists (ACLs) configuration on the EDI gateway allow the EDI data to be changed or deleted.
- Sensitive data that flows unencrypted through the network, enables unauthorized personnel to read them by using network sniffers. This is not just an external threat and can also be perpetrated internally by an employee.
- Improper authentication mechanisms that may result in session hijacking. This can occur both externally at the web-server level and internally at the Oracle or the SQL server levels.
- Loose physical access controls for server cages enable unauthorized entry. This, coupled with the scenario where unused ports on switches are not disabled, can prove disastrous as anyone can enter the network.

It is important to note that almost all of the above threats can be perpetrated from within the enterprise.

Host threats: Host threats in the sample system are as follows:

- Using un-patched servers enables hackers to exploit known vulnerabilities and take control of the server. Viruses and worms also look for such vulnerabilities.
- Unused protocols, ports, and services increase the surface area open to attack – also known as the attack profile.
- Weak authentication methodologies are potential soft spots.
- Lack of clearly defined trust boundaries enable the odd hacker to slip through.
- Improper server hardening guidelines can result in a mismatch between the server configuration and the security context in which it is placed.

Application threats: Application threats encompass exposures due to improper coding practices, lack of understanding of the security requirements, or a holistic view of the application that results in coding for the wrong security context. More than 50 percent of the top 20 SANS/FBI threats are application layer threats. This puts applications, especially those that are highly interconnected like the ones in an EAI system, directly in the line of fire. Key examples of application threats are:

- Code that is prone to buffer overflows, SQL injection or cross-site scripting.
- Defective data encryption resulting in password information or session IDs creeping out along with the free text across the network.
- Trust boundaries that lack clarity and inadequate data validation at the gateway can act as entry points for hackers.

- Intranet rule set being applied in an extranet/ internet scenario poses potential risk.
- Improper configuration of the TIBCO layer and access rights of the application that is using them make security breach possible. Role-based authorization plays a key role here. In an EAI, there are usually several hops between the user and the resource manager (SQL server). There is a high probability of errors occurring at this stage due to improper transfer of privileges. One of the models that can be effective in such cases would be the subsystem resource access model, in which the user identity flows at the application level and not at the operating system level.

nor the network resources to do so. Rating the threats, however, helps organizations determine the amount of resources required for the risk mitigation process. A very popular rating system used for threat analysis is Microsoft's DREAD methodology, in which threat is usually rated against:

Damage Potential: What would be the extent of the damage if the threat were to happen in reality?

Reproducibility: How easy is it to repeat the attack?

Exploitability: How easy is it to launch an attack?

Affected Users: How many users would be affected by the attack?

Discoverability: How easy is it to find the vulnerability?

The above list of parameters can also be

An attack need not always come from the outside; internal stakeholders like employees can perpetrate it from within

A detailed analysis of threats from all the perspectives – network, host and application – gives us the security profile of the application under review. Once all the threats have been enumerated they need to be rated because the degree of counter measures to be taken depends on the rating. There are two formal ways of achieving this. One is to draw up attack trees and the other to draw up attack profiles.

RATE THE THREATS

Once the list of threats is ready, the threats must be rated on the basis of the risk exposure they present to the application architecture. It is, of course, not practical to prepare for all the threats – most organizations have neither the financial

extended to a specific scenario. For example, how easy is it to detect the threat if it were to occur? (In the case of IP Spoofing, for example, it is very difficult to detect the actual fraud until the user detects it.) What is the damage to the image of the organization if the exploit were to be leaked to the public? E.g., if credit card numbers got into the wrong hands the damage to the company's image may be grave.

CONCLUSION

Threat modeling is a continuous process. The threat scenario is dynamic. To prevent a security crisis the analyst has to be right every time, but the hacker needs to be right only once. A few thumb rules to enhance security are:

Follow the principle of least privileges: Processes and users must be given only the minimum amount of privileges required to complete their task.

Follow the policy of defense in depth: Protection and counter measures should not be restricted to the gateway or only at the asset level where breaches are most likely to occur. Preventive processes need to be put in place both upstream and downstream of the vulnerable points.

Do not rely on security by obscurity: Always validate user inputs before sending them to the servers and also ensure a certain amount of server-side validation to reinforce the client-side validation, which is vulnerable to attacks.

Reduce the attack profile: Assume that all the external systems that the application interacts with are insecure. Disable any service, port, or configuration setting that is not in use. This applies to ports on a switch that is currently not in use as well.

Finally, it helps to remember that the security chain is only as strong as the weakest link.

REFERENCES

1. Enterprise Application Integration by William A. Ruh,, Francis X. Maginnis, William J Brown, Wiley; October 2000
2. Securing Internal Networks: The Final Frontier, Mark Bouchard, August 2004, META Group
3. Improving Web Application Security: Threats and Counter measures Roadmap J.D. Meier, Alex Mackman, Michael Dunner, Srinath Vasireddy, Ray Escamilla and Anandha Murukan, Microsoft Corporation, June 2003
4. Information Classification Policy by Security Architecture and Audits group, Infosys Technologies, October 2003
5. British Standard Information Security Management System, Specification with guidance for use BS 7799-2:2002 September 2002
6. Threats and Counter measures, Web Security Threats and Countermeasures, Patterns and Practices, MSDN, Microsoft Corporation. Accessed September 2004
7. Threat Modelling, Patterns and Practices, MSDN Microsoft Corporation, accessed September 2004
8. Twenty Most Critical Internet Security Vulnerabilities, October 2003, SANS Institute, http://www.sans.org/top20/, Accessed September 2004

Author featured in this issue

VISVESWARAN CHIDAMBARAM

Visveswaran Chidambaram is a Principal Architect with the Information System Architecture and Audit Group. He has extensive experience in designing and implementing e-commerce projects for both websites and brick-and-mortar shops that involves EAI as a key component. He is a qualified Chartered account and CISA. He consults on Designing Secure Architecture, Developing Secure Standards, and Ethical Hacking. He can be contacted at crvisveswaran@infosys.com

For information on obtaining additional copies, reprinting or translating articles, and all other correspondence, please contact:

Telephone : 91-80-51173878

Email: SetlabsBriefings@infosys.com