

Quantifying Security Threats and Their Impact

Anis Ben Aissa
Faculty of Sciences of Tunisia
University of Tunis
Tunis, Tunisia

Anis_enit@yahoo.fr

Robert K. Abercrombie,
Frederick T. Sheldon
Oak Ridge National Laboratory
Oak Ridge, TN 37831 USA
+1-865-241-6537/576-1339

abercrombie@ornl.gov
sheldonft@ornl.gov

Ali Mili
College of Computing Sciences
New Jersey Institute of Technology
Newark NJ 07102-1982 USA
+1 973-596-5215

mili@cis.njit.edu

ABSTRACT

In earlier works, we present a computational infrastructure that allows an analyst to estimate the security of a system in terms of the loss that each stakeholder stands to sustain as a result of security breakdowns. In this paper, we illustrate this infrastructure by means of an e-commerce application.

Categories and Subject Descriptors

D.2.8 [Software Engineering]: Metrics; H.4 [Information Systems Applications]: Miscellaneous; K.6.5 [Management of Computing and Information Systems]: Security and Protection

General Terms

Algorithms, Measurement, Performance, Design, Economics, Reliability, Experimentation, Security, Theory, Verification.

Keywords

Cyber Security Metrics, Risk Management, Information Security.

1. INTRODUCTION

Abercrombie et al. [1, 2] present an infrastructure that allows an analyst to estimate the security of a system in terms of the loss that each stakeholder stands to sustain as a result of security breakdowns. The infrastructure in question reflects the values that stakeholders have in each security requirement, the dependency of security requirements on the operation of architectural components, the impact that security threats have on the proper operation of security components, as well as the threat configuration that looms on the operation of the system. This latter feature is the security equivalent of a fault model in reliability modeling.

The submitted manuscript has been authored by a contractor of the U.S. Government under contract DE-AC05-00OR22725. Accordingly, the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CSIRW '09, April 13-15 Oak Ridge, Tennessee, USA
Copyright © 2009 ACM 978-1-60558-518-8-5 ... \$5.00.

We illustrate the proposed quantitative model using a simple example, consisting of an e-commerce application. We illustrate each stakeholder can determine their stake in the secure operation of the system and ensure their individual mission fulfillment. We consider the following criteria:

- The threat configuration under which the application operates, which is represented by a vector of probabilities of occurrence of security breakdowns for a given duration of operation.
- The impact that security breakdowns have on the proper operation of individual components of the architecture (depending on which part of the system each threat targets).
- The dependency that exists between architectural components and security requirements; this relation reflects how/to what extent each component of the system contributes to meeting each security requirement.
- The stakes that each stakeholder has in meeting each clause of the security requirements specification.

In addition to analyzing the e-commerce scenario to quantify security threats and their impact, we discuss how and by whom each aspect of this quantitative model is derived. In this way we can assess the cost effectiveness of security measures, and judiciously allocate the cost of implementing said security measures on the appropriate benefactor.

We build upon the quantitative model of Abercrombie et al [1, 2], and provide a rationale for it (using probability theory) as compared to other methods [1, 3] in Section 2. In sections 2, 3, and 4, we discuss in turn the stakes matrix, the dependency matrix, and the impact matrix; in section 5, this analysis is completed by a discussion of the threat configuration.

2. ILLUSTRATION: AN E-COMMERCE APPLICATION

We illustrate the use of our Cyber Security Econometrics Model [1, 2] on a practical application, namely an e-commerce system. We derive in turn the three matrices of interest, starting with the stakes matrix. To this effect, we first identify the security requirements [4], then the stakeholders and their stakes in meeting these requirements.

2.1 Security Requirements

We identify the following security requirements for this application [5, 6]:

- **Confidentiality.** Need to ensure that data are accessible only to those who are authorized to view them [7].
- **Integrity.** Need to ensure that information that is displayed or transmitted had not been altered by unauthorized parties.
- **Availability.** Need to ensure that the e-commerce application is operational whenever a customer needs to use it.
- **Non-repudiation.** Need to ensure that no party in an operation can deny participating in the operation.
- **Authenticity.** Need to ensure that all actors in a system are properly authenticated, and their privileges and responsibilities defined accordingly.
- **Privacy.** Need to ensure that information pertaining to system users is not improperly divulged.

2.2 Stakes and Stakeholders – The Stakes Matrix

We recognize four stakeholders in this application, namely: the customer, the merchant, the technical intermediary, and the financial intermediary. We briefly review the stakes that they have in meeting the security requirements, as these determine the corresponding values in the stakes matrix.

- **The Customer.** The stakes that the customer has in the secure operation of the system include: the loss of confidential information, which the customer may provide during the e-commerce transaction; transaction failure; identity theft.

meet each security requirement), we produce the following stakes matrix, as show in Table 1.

3. THE DEPENDABILITY MATRIX

The dependability matrix represents how (to what extent) security requirements are dependent on the proper operation of system components. In order to derive this matrix, we must first look at the architecture of the application. We adopt the following tiered architecture, borrowed from [8, 9]. We review in turn several of the tiered components of this architecture: Web browser, proxy servers, Web servers, application servers, and database servers.

3.1 Tiered Components of Dependability Matrix

The following sub-sections (3.1.1. through 3.1.5) follow the logic developed by Ahmed for the eBay e-commerce scalability scenario [9].

3.1.1 Web Browser

The end user typically interacts with the Website through a Web browser. Web browsers support user interface modifiability in a wide variety of ways, as the user interface that the browser supports is not hardwired but it is specified via HTML.

3.1.2 Proxy servers

Request from individual browsers may first arrive at a proxy server, which exists to improve the performance of the Web-based

Table 1. Stakes (ST) Matrix: Cost of failing a security requirement stakes in \$/Hour

ST		Security Requirements					
		Confidentiality	Integrity	Availability	Non-repudiation	Authenticity	Privacy
Stakeholders	Customer	10	5	3	4	6	12
	Merchant	120	70	140	110	105	6
	Tech Int	20	20	40	20	30	20
	Fin Int	20	60	50	40	40	60

- **The Merchant.** The stakes that the merchant has in the secure operation of the system include: the loss of business that may result from failing the availability requirement; the loss of customer loyalty that may result from failing the availability requirement; the loss of customer loyalty that may result from failing the confidentiality or the privacy requirements; the loss of business that may result from failing the integrity requirement, etc.
- **The Technical Intermediary.** The stakes that the technical intermediary has in the secure operation of the system include: the loss of business from the merchant; the loss of reputation for good service, which may result in lost corporate value.
- **The Financial Intermediary.** The stakes that the financial intermediary has in the secure operation of the system include: financial losses that result from malicious activities by customers; the loss of business from the merchant; the loss of reputation for good service, which may result in lost corporate value.

Based on a quantification of these stakes in terms of dollars per hours of operation (under the hypothesis that the system fails to

system. The proxy server cache frequently accessed Web pages so that users may retrieve them without having to access the main Website [9]. However, if a user chooses a particular item, with the intention of bidding or selling, then he must be shown real-time data. These proxy servers are typically located close to the users, often on the same intra-network, thus saving a tremendous amount of communication and computation resources.

3.1.3 Web servers

The HTTP or HTTPS request reaches the Web server. The Web servers are multithreaded, utilizing a pool of threads, each of which can be dispatched to handle an incoming request. A multithreaded server is less susceptible to bottlenecks (and hence long latency) when a number of long-running HTTP or HTTPS requests (such as credit card validation) arrive because other threads in the pool are still available to serve incoming requests [9, 10]. This introduces concurrency at the Web server level. Upon analyzing the request, the Web server sends it to an application server that responds using the service of a database.

3.1.4 Application servers

From the Web server the request is forwarded to an application server. These application servers run in the middle business rules and application architecture as illustrated in the figure above. These servers implement business logic and connectivity, which dictate how clients and servers interact. This allows the databases to concentrate on the storage, retrieval, and analysis of data without worrying about precisely how that data will be used.

3.1.5 Database servers

Finally, the request for service arrives at the database, where it is converted into an instruction to add, modify, or retrieve information. The relation database management system (RDBMS) must be able to support all incoming requests from the application servers.

3.2 Generation of the Dependency Matrix

The question we address in this section is how to estimate the probability that a particular security requirement is violated in the course of operating the system for some period of time [1, 2, 11]. The idea that we pursue here is to link the probability of failing a particular requirement with the probability of failure of a component of the system. The elucidation of this probabilistic link involves an analysis of the system's architecture, to determine which component contributes to meeting which requirement.

Assuming that separate commercial components of the same type play interchangeable roles, we do not need to represent individual components in the dependability matrix; it suffices to represent general families of components. Hence we must consider the following (families of) components:

- Browser,
- Proxy Server,
- Router/ Firewall,
- Load Balancer,
- Web Server,
- Application Server, and
- Database Server.

Assuming no more than one component fails at a time, and considering the additional event that no component has failed, the dependability matrix has $(7+1=)$ 8 columns and 6 rows (one for each security requirement), for a total of 48 entries. We cannot comment on all 48 entries, but will give below a sample of the

reasoning that goes into filling the dependability matrix; those values on which we comment are represented in Table 2.

- If no component fails, then (presumably) all security requirements are satisfied.
- If one of the database components fails, then this does not affect the availability of the system (since according to our hypothesis, the other database server is necessarily operational); loss of a database server may affect response time, but not necessarily availability.
- Assuming confidential information is stored in only one database (for enhanced protection), then failure of a database server causes a failure with respect to confidentiality, authentication and privacy with probability 0.5.
- If a Browser fails then availability is not satisfied.
- If a Proxy server fails, then availability is not satisfied.
- If the Router/Firewall fails, then no dimension of security is satisfied.
- If a Web server fails then all the dimensions of security have probability 0.33 to fail (all the queries that are routed to that server lead to unpredictable outcomes).
- If the router is assumed to check when a Web server fails, then these probabilities would be 0.0.

4. THE IMPACT MATRIX

The impact matrix relates component failures to security threats; specifically, it represents the probability of failure of components given that some security threat (from a pre-catalogued set) has materialized. The first step in deriving the impact matrix is, of course, the derivation of the set of threats that we wish to consider; this is akin to defining a fault model (including a set of possible faults) in the analysis of the reliability of a system.

4.1 Threats on communication protocols

This threat category exploits the weaknesses in the basic Internet protocols such as TCP/IP, HTTP, and FTP. The main attack modes include:

- Attacks to disable Web services (e.g., DoS, Denial of Information [DoI]),
- Eavesdropping (man-in-the-middle) communications attacks,
- The replacement and the manipulation of data, and
- Covert channel, data exfiltration and diversion of protocols.

Table 2. *Dependency (DP) Matrix: Links requirements with components*

DP		Components							
		Browser	Proxy Server	Router/ Firewall	Load Balancer	Web Server	Appl. Server	Database Server	No Failure
Security Requirements	Conf	0.2	0.2	1.0	1.0	0.333	0.333	0.5	0.0
	Int	0.2	0.2	1.0	1.0	0.333	0.333	0.0	0.0
	Avail	1.0	1.0	1.0	1.0	0.333	0.333	0.0	0.0
	NR	0.2	0.2	1.0	1.0	0.333	0.333	0.0	0.0
	Auth	0.2	0.2	1.0	1.0	0.333	0.333	0.5	0.0
	Priv	0.2	0.2	1.0	1.0	0.333	0.333	0.5	0.0

4.2 Threats on systems

This category includes the attacks which exploit the weaknesses at the level of the standard applications of the server. This problem is supported by the standardization of operating systems (UNIX, NT, XP, or Vista) and standard applications of communication (SMTP e-mailer, browser using HTTP or still use of SQL for databases). The different possibilities of attacks included in this category are:

- Attacks on unused network services and not or weakly protected;
- Attacks on the availability of the service by use of the bugs in applications; and
- Attacks aiming at accessing the computer systems of the company.

4.3 Threats on the information

This last threat type can be used to exploit brand recognition for profit. Another example would deface (i.e., introduce false information) a site to maliciously affect the brand image of a company. In general, attack forms include:

- Attacks against a site by manipulation of publically available Internet information resulting in disinformation, defrauding customers, compromising the integrity of the company,
- Attacks resulting in attaining information illegally from a site via secretive means (e.g., infiltration via social engineering techniques of phishing, spear-phishing and whaling) and
- The modifications of contents via transactions resulting in data exfiltration (e.g., credit card numbers and other personally identifiable information [PII], and identity theft).

4.4 The passive listening

The current mechanism used by attackers is monitor (listen to) communication traffic on the network to try to obtain information concerning authentication such as the login and the password of a user. Once this obtained, the attacker uses this information to connect to the server and impersonate the real user or install a sniffing backdoor tool (non-promiscuous or promiscuous) to repeatedly access the machine.

4.5 Virus

The infection of the server by a virus can provoke total or partial unavailability, but more serious still is the fact that the server can propagate the virus to system users. For example, the Asprox virus infection weaves a complex chain of dependencies involving bots that perform SQL injection on vulnerable Web servers and visitors whose machines get compromised simply by visiting infected Websites [12].

4.6 Trojan

The Trojan horse, also known as Trojan, in the context of computing and software, describes a class of computer threats that appears to perform a desirable function but in fact performs undisclosed malicious functions that allow unauthorized access to the host machine, giving them the ability to save their files on the user's computer or even watch the user's screen and control the computer. Trojan horse payloads are almost always designed to cause harm, but can also be harmless. They are classified based on how they breach and damage systems. The main types of Trojan horse payloads are:

- Remote Access,
- Data Destruction,
- Downloader/dropper,
- Server Trojan (Proxy, FTP, IRC, Email, HTTP/HTTPS, etc.), and
- Disable security software.

4.7 Denial-of-Service (DoS)/Distributed DoS

A denial-of-service attack (DoS attack) or distributed denial-of-service attack (DDoS attack) is an attempt to make a computer resource unavailable to its intended users. A DoS attack can be perpetrated in a number of ways. The five basic types of attack are:

- Consumption of computational resources, such as bandwidth, disk space, or processor time,
- Disruption of configuration information, such as routing information,
- Disruption of state information, such as unsolicited resetting of TCP sessions,
- Disruption of physical network components, and

Table 3. *Impact (IM) Matrix: Links components to threats*

IM		Threats								
		Comm	Sys	Info	List	Virus	Troj	DoS	DB	NoT
Components	Brws	0.0	0.1	0.1	0.1	0.3	0.4	0.2	0.0	0.0
	Prox	0.5	0.1	0.1	0.3	0.3	0.4	0.2	0.0	0.0
	R/FW	0.5	0.1	0.1	0.3	0.3	0.4	0.6	0.0	0.0
	LB	0.0	0.1	0.1	0.1	0.3	0.4	0.6	0.0	0.0
	WS	0.0	0.6	0.6	0.2	0.3	0.4	0.2	0.0	0.0
	AS	0.0	0.1	0.1	0.1	0.3	0.4	0.2	0.0	0.0
	DBS	0.0	0.1	0.1	0.0	0.5	0.6	0.3	0.8	0.0
	NoF	0.4	0.3	0.1	0.1	0.05	0.05	0.1	0.2	1.0

- Obstructing the communication media between the intended users and the victim so that they can no longer communicate adequately.

4.8 Threats on the database

One of the possible attacks, in principle, modifies, indirectly the SQL statement sent to the server, by including special character strings instead of the parameters that are expected by the application software. If a web application fails to properly check user-supplied input, it is possible for an attacker to alter the construction of the a SQL statement. This technique allows getting back confidential information of the database or simply meta given. We can make, for example, a normal call in the address bar of the browser: `http: // server / prog? User=name_user`. We can then make a call falsified by the type: `http: // server / prog? User=other_user`. Either we obtain directly information concerning the other user, or an error which can give us indications that allow learning, for example, as to names of users. Worse yet, if the attacker is able to modify a SQL statement, the process could run with the same permissions as the original statement resulting in the attacker possibly gaining control of the database.

4.9 Generating the Impact Matrix

Given that we have cataloged eight security threats, the impact matrix will have nine columns, one for each threat plus one for the absence of threats. On the other hand, it has 8 rows, one for each component plus one for the event that no component has failed during the unitary time period. This gives a total of 72 entries; we will comment on some of them, which we will represent in the Table 3.

The absence of threats does not cause the failure of any component, and leads to event NoF (No Failure) with probability 1.0.

- We estimate that threats to the database cause a failure of the database with probability 0.8, say, to make provisions for the case where an attack fails to achieve its goal; they may cause event NoF (No Failure) with probability 0.2. We assume that because the DB component is the only target of this threat, the probability that it causes a failure of any other component is 0.0.

Table 4. Vector PT providing threat probability

PT		Probability
Threats	Comm	0.01
	Sys	0.02
	Info	0.01
	List	0.01
	Virus	0.03
	Troj	0.06
	DoS	0.03
	DB	0.02
	NoT	0.81

- Generally, the row labeled NoF represents the probability of failure of each threat, i.e. the probability that it does not cause any component to fail.
- The threat on communication protocol (Comm) targets the proxy servers and the routers; we assume that the probability that it causes a failure of the other components is 0.0.
- A virus has some likelihood of affecting any component of the system, through propagation.
- A Trojan horse has some likelihood of targeting any component of the system, through propagation.
- The threat passive listening (list) targets primarily the components that are involved with communication.
- The denial of service attacks (DoS) may target the bottlenecks of the architecture, for maximal effect.

5. THREAT CONFIGURATION

Vector PT (Table 4) characterizes the threat situation by assigning to each threat category the probability that this threat will materialize over a unitary period of operation (say, an hour). We assume that no more than one threat can materialize within a unitary period of time, and we make provisions for the case where no threat materializes. Hence this vector contains a probability distribution of complementary events. We assume that, in light of “log” data, known vulnerabilities, and known perpetrator behavior, that we can determine that the prevailing threats have the probabilities indicated below.

Using this data, we now compute the vector of mean failure costs [2, 13, 14], using the formula

$$MFC = ST \circ DP \circ IM \circ PT.$$

Substituting each matrix by its value, in Table 5, we find:

Table 5. Stakeholder Mean Failure Cost (MFC)

Stakeholders	MFC \$/hour
Customer	8.11
Merchant	112.97
Technical intermediary	31.17
Financial intermediary	54.24

6. CONCLUSIONS

In this paper we have illustrated the application of the CSES/MFC model for estimating system security by means of a concrete example. The quantification of security attributes by means of costs to stakeholders opens a wide range of possibilities for further economics based analysis, and provides a valuable resource for rational decision making (e.g., risk mitigation planning). Our future plans call for exploring such opportunities.

7. REFERENCES

- [1] R. K. Abercrombie, F. T. Sheldon, and A. Mili, “Synopsis of Evaluating Security Controls Based on Key Performance Indicators and Stakeholder Mission Value,” in 11th IEEE High Assurance Systems Engineering Symposium (HASE '08), Nanjing, China, 2008, pp. 479-482.
- [2] F. T. Sheldon, R. K. Abercrombie, and A. Mili, “Methodology for Evaluating Security Controls Based on

- Key Performance Indicators and Stakeholder Mission,” in Proceedings of 42nd Annual Hawaii International Conference on System Sciences (HICSS-42), Waikoloa, HI, 2009, pp. 10.
- [3] R. K. Abercrombie, F. T. Sheldon, and A. Mili, “Managing Complex IT Security Process with Valued Based Measures,” in 2009 IEEE Symposium on Computational Intelligence in Cyber Security (CICS 2009), Nashville, TN, 2009, pp. 7.
 - [4] D. Firesmith, “Specifying Reusable Security Requirements,” *Journal of Object Technology*, vol. 3, no. 1, pp. 61-75, 2004.
 - [5] S. V. d. Rocha, Z. Abdelouahab, and E. Freire, “Requirement Elicitation Based on Goals with Security and Privacy Policies in Electronic Commerce,” in *Anais do WER05 - Workshop em Engenharia de Requisitos*, Porto, Portugal, 2005, pp. 63-74.
 - [6] K. C. Sekaran, “Requirements Driven Multiple View Paradigm for Developing Security Architecture,” in *PWASET, Proceedings of World Academy of Science, Engineering and Technology*, 2007, pp. 156-159.
 - [7] V. D. Sawma, and R. L. Probert, “E-Commerce Authentication: An Effective Countermeasures Design Model,” in *ICEIS 2003, Proceedings of the 5th International Conference on Enterprise Information Systems*, Angers, France, 2003, pp. 447-455.
 - [8] D. Pritchett, “The eBay Architecture: Striking a Balance Between Stability, Feature Velocity, Performance, and Cost,” in *Colorado Software Summit 2007*, Keystone, CO, 2007, pp. 1-39.
 - [9] M. U. Ahmed, “eBay - eCommerce Platform, A Case Study in Scalability,” McGill University, pp. 1-13.
 - [10] Ranjit Goswami, S. K. De, and B. Datta, “E-business Adoption in Select Indian Firms and Segments: A Stakeholders’ Approach through Select Indian Portals Analysis,” in *CISTM 2005, Conference of Information Science, Technology and Management 2005*, pp. 1-22.
 - [11] F. T. Sheldon, R. K. Abercrombie, and A. Mili, “Evaluating Security Controls Based on Key Performance Indicators and Stakeholder Mission,” in *Proceedings of the 4th Annual Cyber Security and Information Intelligence Research Workshop*, Oak Ridge, TN, 2008.
 - [12] Y. Shin, S. Myers and M. Gupta, A Case Study on Asprox Infection Dynamics, *Sixth Conference on Detection of Intrusions and Malware & Vulnerability Assessment (DIMVA 2009)*, Springer-Verlag Berlin Heidelberg, Milan, Italy, 2009, pp. 1–20.
 - [13] A. Mili, and F. T. Sheldon, “Measuring Reliability as a Mean Failure Cost,” in *Proceedings of the 10th IEEE High Assurance Systems Engineering Symposium*, Dallas, TX, 2007, pp. 403-404.
 - [14] A. Mili, and F. T. Sheldon, “Challenging the Mean Time to Failure: Measuring Dependability as a Mean Failure Cost,” in *Proceedings of 42nd Hawaii International Conference on System Sciences (HICSS-42)*, Waikoloa, HI, 2009, pp. 10.



FACULTY OF SCIENCES OF TUNIS, TUNISIA

Quantifying Security Threats and Their Impact

Anis Ben Aissa
Faculty of Sciences of Tunisia
anis_enit@yahoo.fr

Robert K. Abercrombie
Oak Ridge National Laboratory
abercrombie@ornl.gov

Frederick T. Sheldon
Oak Ridge National Laboratory
sheldonft@ornl.gov

Ali Mili
New Jersey Institute of Technology
mili@cis.njit.edu

Outline



Introduction



A cascade of linear models



Illustration: E-Commerce Application



Summary

Introduction

Abercrombie et al presented an infrastructure that allows an analyst to estimate the security of a system in terms of the loss that each stakeholder stands as a result of security breakdowns.

The infrastructure reflects:

- The stakes that stakeholders have in each requirement.
- The dependency of security requirements on the operation of architecture components.
- The impact that security threats have on the proper operation of security components.
- The threat configuration that looms on the operation of the system.

Outline

- Introduction**
- A cascade of linear models**
- Illustration: E-Commerce Application**
- Conclusion**

Introduction	A Cascade for Linear Models	Illustration: E-Commerce Application	Summary
	The Stakes Matrix	The Dependency Matrix	The Impact Matrix
			Summary

A Cascade for Linear Models

- We consider a system S we let H_1, H_2, \dots, H_K be stakeholders of the system
- We let R_1, R_2, \dots, R_n be security requirements that we wish to impose on the system
- We let PR_j be the probability that the system fails to meet security requirement R_j
- We let MFC_i (Mean Failure Cost) represent the cost to stakeholder H_i that may result from a security failure.

The mean failure cost for stakeholder T_i can be written as:

$$MFC_i = \sum_{1 \leq j \leq n} ST_{i,j} \times PR_j.$$

Introduction	A Cascade for Linear Models	Illustration: E-Commerce Application	Summary
	The Stakes Matrix	The Dependency Matrix	The Impact Matrix
			Summary

A Cascade for Linear Models


- If we let the MFC be the column-vector of size K that presents mean failure costs
- Let ST be the $K \times n$ matrix that presents stakes, and let PR be the column-vector of size n that represents probability of failing security requirements
- Using the matrix product (\circ) we can write :

$$MFC = ST \circ PR$$

Introduction	A Cascade for Linear Models	Illustration: E-Commerce Application	Summary
	The Stakes Matrix	The Dependency Matrix	The Impact Matrix
			Summary

A Cascade for Linear Models

- ✚ We consider the architecture of system S, and let C_1, C_2, \dots, C_h be the components of system S.
- ✚ The security requirements depends on which component of the system is operational.
- ✚ Let F_j denote the event that the system fails with respect to requirement R_j
- ✚ We can write the probability of failure with respect to R_j :




$$PR_j = \sum_{k=1}^{m+1} P(F_j | E_k) \times P(E_k).$$

Introduction	A Cascade for Linear Models	Illustration: E-Commerce Application	Summary
	The Stakes Matrix	The Dependency Matrix	The Impact Matrix
			Summary

A Cascade for Linear Models

- ✚ If we introduce the DP (Dependency) matrix which has n rows and h+1 columns, and where the entry at row j and column k is the probability that the system fails with respect to security requirement j given that component k has failed.
- ✚ We introduce vector PE of size h+1, such that PE_k is the probability of event E_k .
- ✚ We can write:



$$PR = DP \circ PE.$$

Introduction	A Cascade for Linear Models	Illustration: E-Commerce Application	Summary
	The Stakes Matrix	The Dependency Matrix	The Impact Matrix
			Summary

A Cascade for Linear Models

- ✚ Components of the architecture may fail to operate properly as a result of security breakdowns brought about by malicious activity.
- ✚ In order to continue the analysis, we must specify the catalog of threats
- ✚ Let T_1, T_2, \dots, T_p , represent the event that a cataloged threat has materialized, and we let T_{p+1} , be the event that no threat has materialized.
- ✚ Let PT be the vector of size $p+1$, We can write:

$$PE_k = \sum_{q=1}^{p+1} P(E_k | T_q) \times PT_q.$$

Introduction	A Cascade for Linear Models	Illustration: E-Commerce Application	Summary
	The Stakes Matrix	The Dependency Matrix	The Impact Matrix
			Summary


A Cascade for Linear Models


- ✚ If we introduce the IM (Impact) matrix, which has $h+1$ rows and $p+1$ columns, and where the entry at row k and column q is the probability that component C_k fails given that threat q has materialized
- ✚ We introduce vector PT of size $p+1$, such that PT_q is the probability of event T_q ,
- ✚ Then we can write:


$$PE = IM \circ PT$$

Introduction	A Cascade for Linear Models	Illustration: E-Commerce Application	Summary
	The Stakes Matrix	The Dependency Matrix	The Impact Matrix
			Summary

A Cascade for Linear Models





 Given the stakes matrix ST, the dependability matrix DP , the impact matrix IM and the threat vector PT.

 We can derive the vector of mean failure costs by the following formula:



$$\text{MFC} = \text{ST} \circ \text{DP} \circ \text{IM} \circ \text{PT}$$

Outline

-  Introduction
-  A cascade of linear models
-  **Illustration: E-Commerce Application**
-  Summary

E-Commerce Application



Stakeholders:

- The Customer
- The Merchant
- The Technical Intermediary
- The Financial Intermediary



Security Requirements:

- Confidentiality
- Integrity
- Availability
- Non-repudiation
- Authentication
- Privacy

Table 1. Stakes (ST) Matrix: Cost of failing a security requirement stakes in \$/Hour

ST		Security Requirements					
		Confidentiality	Integrity	Availability	Non - repudiation	Authenticity	Privacy
Stakeholders	Customer	10	5	3	4	6	12
	Merchant	120	70	140	110	105	6
	Tech Int	20	20	40	20	30	20
	Fin Int	20	60	50	40	40	60

E-Commerce Application



In order to derive the Dependability matrix we need the architecture of the application Fig1.

Components:

- ✓ Web Browser
- ✓ Proxy Servers
- ✓ Router/Firewall
- ✓ Web Servers
- ✓ Application Servers
- ✓ Database Servers

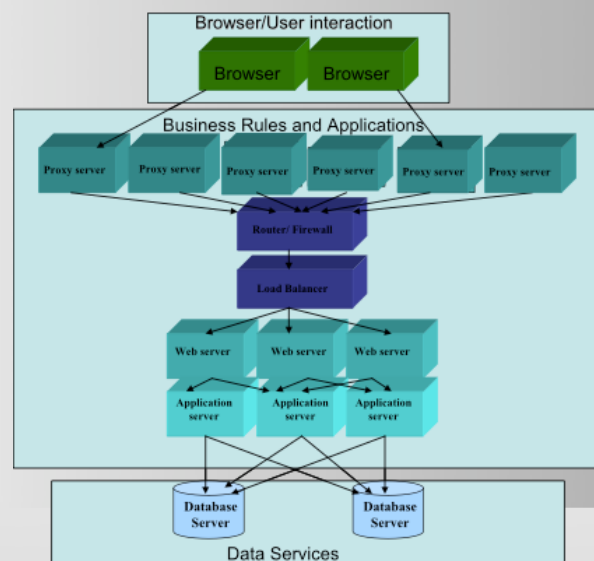


Fig1. Subsystems in the e-commerce architecture

Introduction		A Cascade for Linear Models				Illustration: E-Commerce Application			Summary
--------------	--	-----------------------------	--	--	--	--------------------------------------	--	--	---------

E-Commerce Application

Dependency (DP) Matrix: Links requirements with components

DP		Components							
		Browser	Proxy server	Router/Firewall	Load Balancer	Web Server	Appl. Server	Database Server	No Failure
Security Requirements	Confidentiality	0.2	0.2	1.0	1.0	0.333	0.333	0.5	0.0
	Integrity	0.2	0.2	1.0	1.0	0.333	0.333	0.0	0.0
	Availability	1.0	1.0	1.0	1.0	0.333	0.333	0.0	0.0
	Non -repudiation	0.2	0.2	1.0	1.0	0.333	0.333	0.0	0.0
	Authenticity	0.2	0.2	1.0	1.0	0.333	0.333	0.5	0.0
	Privacy	0.2	0.2	1.0	1.0	0.333	0.333	0.5	0.0

Introduction		A Cascade for Linear Models				Illustration: E-Commerce Application			Summary
--------------	--	-----------------------------	--	--	--	--------------------------------------	--	--	---------

E-Commerce application

✚ The impact matrix relates component failures to security threats: it represents the probability of failure of components given that some security threat has materialized.

✚ The first step in deriving the impact matrix is the derivation of the set of threats that we wish to consider; this is akin to defining a fault model.

- ✓ Threats on communication protocols
- ✓ Threats on the systems and the standars applications
- ✓ Threats on the information
- ✓ The passive listening
- ✓ Virus
- ✓ Trojan
- ✓ Denial of service
- ✓ Threats on the database

Introduction	A Cascade for Linear Models	Illustration: E-Commerce Application	Summary
--------------	-----------------------------	--------------------------------------	---------


E-Commerce application

Impact (IM) Matrix: Links components to threats

IM		Threats								
		Comm	Sys	Info	List	Virus	Troj	DoS	DB	NoT
Components	Brws	0.0	0.1	0.1	0.1	0.3	0.4	0.2	0.0	0.0
	Prox	0.5	0.1	0.1	0.3	0.3	0.4	0.2	0.0	0.0
	R/FW	0.5	0.1	0.1	0.3	0.3	0.4	0.6	0.0	0.0
	LB	0.0	0.1	0.1	0.1	0.3	0.4	0.6	0.0	0.0
	WS	0.0	0.6	0.6	0.2	0.3	0.4	0.2	0.0	0.0
	AS	0.0	0.1	0.1	0.1	0.3	0.4	0.2	0.0	0.0
	DBS	0.0	0.1	0.1	0.0	0.5	0.6	0.3	0.8	0.0
	NoF	0.4	0.3	0.3	0.1	0.05	0.05	0.1	0.2	1/0

Introduction	A Cascade for Linear Models	Illustration: E-Commerce Application	Summary
--------------	-----------------------------	--------------------------------------	---------

E-Commerce application

 Threat Configuration: Vector PT characterizes the threat situation by assigning to each category of threats the probability that this threat will materialize over a unitary period of operation

PT		Probability
Threats	Comm	0.01
	Sys	0.02
	Info	0.01
	List	0.01
	Virus	0.03
	Troj	0.06
	DoS	0.03
	DB	0.02
	NoT	0.81

E-Commerce application

✚ Using this data, we can compute the vector of mean failure cost using the formula: $MFC = ST \circ DP \circ IM \circ PT$.

Stakeholders	Mean failure cost \$/hour
Customer	8.11
Merchant	112.97
Technical intermediary	31.17
Financial intermediary	54.24

Outline

- ✚ Introduction
- ✚ A cascade of linear models
- ✚ Illustration: E-Commerce Application
- ✚ Summary

Summary

- ✓ we have illustrated the application of the CSES/MFC model for estimating system security by means of a concrete example.
- ✓ The quantification of security attributes by means of costs to stakeholders opens a wide range of possibilities for further economics based analysis, and provides a valuable resource for rational decision making.
- ✓ Our future plans call for exploring such opportunities.

Thank you