



# Information assurance metric development framework for electronic bill presentment and payment systems using transaction and workflow analysis

G.B. Tanna<sup>a</sup>, M. Gupta<sup>a</sup>, H.R. Rao<sup>b,\*</sup>, S. Upadhyaya<sup>c</sup>

<sup>a</sup>*M&T Bank, United States*

<sup>b</sup>*MSS Department, School of Management, SUNY Buffalo, United States*

<sup>c</sup>*CSE Department, College of Engineering, SUNY Buffalo, United States*

Received 31 August 2003; received in revised form 30 June 2004; accepted 30 June 2004

Available online 1 September 2004

## Abstract

One of the fastest growing applications in the banking arena is Electronic Bill Presentation and Payment (EBPP), driven primarily by a desire to reduce costs associated with issuing and settling physical bills. EBPP is a secure system for companies to electronically present bills and other related information to their customers, and host the secure payment of these bills. This paper puts forth information assurance issues that are analyzed from a workflow and transaction analysis perspective. Various aspects and technologies deployed in EBPP systems are discussed with a view to understand security underpinnings. The paper develops a framework for the measurement of security levels of any EBPP system, which will help security personnel to ensure a higher level of understanding of information assurance issues and proactively engage in elevating security measures and fraud protection in their organizations. A step-by-step procedure is developed to help IT security managers and administrators to understand the metrics that can define proactive and reactive security service delivery levels, and implement the measurement framework that is necessary to demonstrate performance against these metrics.

© 2004 Elsevier B.V. All rights reserved.

*Keywords:* EBPP systems; Electronic payment and presentation; Information assurance; Security measurement; Security metrics; Threat analysis; Workflow analysis

## 1. Introduction

Advances in telecommunications and information technologies are leading to a revolution in the payment industry in the banking arena. With the rise in the use of Internet for carrying out operations and

\* Corresponding author. Tel.: +1 716 636 8866; fax: +1 716 636 6117.

*E-mail address:* [mgmtrao@buffalo.edu](mailto:mgmtrao@buffalo.edu) (H.R. Rao).

functions that were traditionally done through paper, pen and snail-mail, with the help of electronic systems, the importance of Electronic Bill Presentation and Payment (EBPP) cannot be over-emphasized. So, although paper-checks are still the dominant way of making payments [5], EBPP is a modern and convenient form of making payments with the expediency of instant confirmation and faster postings of those payments [6].

Notably, there are a couple of payment services models: (1) The Biller-Centric Model or the Direct Model, and (2) The Payer-Centric Model or the Consolidator Model. However, the value proposition inherent in payer-centric systems is more balanced than the traditional method of posting invoices on billers' websites. Biller-centric solutions put the burden on the payer to visit multiple sites for their invoices, and there is little incentive for payers—particularly large organizations—to want to do this [7].

In this paper, we focus our attention on the payer-centric model. This model of the bill-payment process involves five parties or entities. They are: (1) Consumer, (2) Consumer's Financial Institution, (3) Biller, (4) Biller's Financial Institution and (5) The Payment Network. A typical flow of information between the five parties/entities involved is depicted in Fig. 1.

The model presented in the paper can aid IS managers to understand their organization's IS posture and manage IS risks by taking proactive measures rather than reactive ones. The contribution of the paper is twofold—it develops (1) detailed transactional workflows that can be used to expose the vulnerabilities that are prevalent in the electronic bill-

payment system using a comprehensive threat analysis model (details of which are discussed in a later section) to conduct a systematic threat analysis; (2) a framework and a step-by-step procedure to help IT security managers and administrators to develop a vulnerability score to understand the metrics that can define proactive and reactive security service delivery levels, and implement the measurement capability that is necessary to demonstrate performance against these metrics. This can serve as launching pad for risk assessment of EBPP or similar systems. Since, some parts of the model are based on subjective evaluations of risk factors, experience of the reviewer, and biased outlook towards the environment, etc., would come into play. Even so, the framework presented here can be adapted to the environment and be refined subsequent to feedback-based iterative corrections. The score thus derived can serve as baseline for assessing the strength of a system's security posture with time and improvements.

The rest of the paper is organized as follows. Section 2 discusses the technologies and components of the EBPP systems. A detailed analysis of the architecture is also carried to examine information assurance issues in the links. Section 3 presents a state diagram and transaction analysis of EBPP systems with respect to security issues. Then we present an extension of the STRIDE<sup>1</sup> threat analysis model and its detailed adaptation with the EBPP system in Section 4. Section 5 outlays threat identification and analysis of EBPP system vis-à-vis the adapted STRIDE Model. Identification and measurement of security of the system is carried out through the transaction workflow analysis of the EBPP system. Finally, a framework, based on workflow states, to derive an overall vulnerability score for the EBPP system is developed in Section 6. This security measurement framework will help IT security managers and administrators gauge the security posture of the EBPP system and proactively engage in remediation of the weaknesses and IS risk mitigation in general.

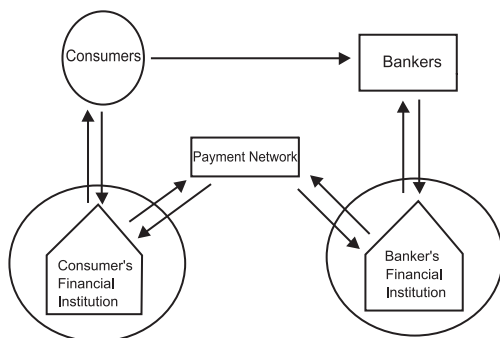


Fig. 1. A typical information flow for a bill-payment system [1].

<sup>1</sup> Developed by Microsoft, the STRIDE Model gets its name from the first letter of the threat categories identified by them, e.g., S=Spoofing Identity, T=Tampering of Data and so on.)

## 2. Electronic Bill Presentation and Payment systems—components and architectures

The bill-payment system for a typical financial institution is hosted and managed by the technology division of its operations. This is not unusual for any commercial bank. Illustrated below is the system architecture of an EBPP system of a typical financial institution.

### 2.1. Architecture overview

As shown in Fig. 2, commercially available EBPP system architectures are mainly comprised of the following main tiers [11]:

- (a) *Presentation Tier*: This tier consists of all the devices and techniques that are used to communicate with the application (Fig. 2: Component A) (see also Table 1). It basically comprises desktop-based browsers that use HTTP or SSL and mobile devices that use WAP or WTLS for communication. Additionally, interfaces also exist for back-end communication that either use TCP/IP, X.25, SNA and/or 3270 (Fig. 2: Component: K). In a typical scenario, after EBPP session initiation, the encrypted (SSL) packets, from the presentation tier of discussed architecture, will leave Data Terminal Equipment (DTE) of the user premises and will go through a number of Data Communication Equipments (DCEs) before hitting the DTE at the host of EBPP system. This WAN-based communication could be leased line, circuit-switched (ISDN) or packet-switched as is also discussed in Section 2.4. The complete EBPP transaction processing after firewall in Fig. 2 is assumed to be on private network and does not use the Internet. Details of the transaction path before border router or firewall of the host of EBPP system is not included in the security assessment of the EBPP system.
- (b) *Application Tier*: This tier consists of various sub-components like the Web server, Application Server or the business integration software that hosts the business logic. The Web server (Fig. 2: Component D, e.g., Weblogic, Apache, etc.) provides a platform to run the application

server. A typical application server and business integration platform includes IBM's Websphere, which is very popular nowadays (Fig. 2: Component J).

- (c) *Database Tier*: The application layer then communicates with the database (Component G) through SQL over sockets for storing and retrieving of customer data. Typical databases that are found in large-scale EBPP systems include Sybase, Oracle 8i and above, Main-frame installations like Tandem with NON-STOP SQL are highly common.
- (d) *Other Components*: Some other components that are included in the architecture but do not belong to a specific tier are the Internet (Component: B), Firewall (Component: C) for security, The LDAP services for authentication (Component: E), Transaction Processors (Component: H) and the host servers (Component: I).

### 2.2. Protocols and components used

Various protocols that are used in transmitting this information across the entire system spanning various sub-systems are listed in Table 2. The nodes have been labeled to illustrate the proper sub-system.

### 2.3. Network standards and technologies deployed

Every request from the customer's Web browser is protected by an SSL-channel as it travels across the Internet. SSL provides 128-bit encryption for all traffic, ensuring the confidentiality and integrity of the data when it arrives at the Web server, which is a necessity to comply with National Automated Clearing House Association (NACHA)<sup>2</sup> standards.

All traffic over the network passes through a firewall that allows only HTTPS Web traffic by only keeping those ports open and blocks all other non-HTTP-based communication. Also, the network is

<sup>2</sup> NACHA is the leading organization in developing electronic solutions to improve the payments system and represents more than 12,000 financial institutions through direct memberships and a network of regional payments associations, and 650 organizations through its industry councils.

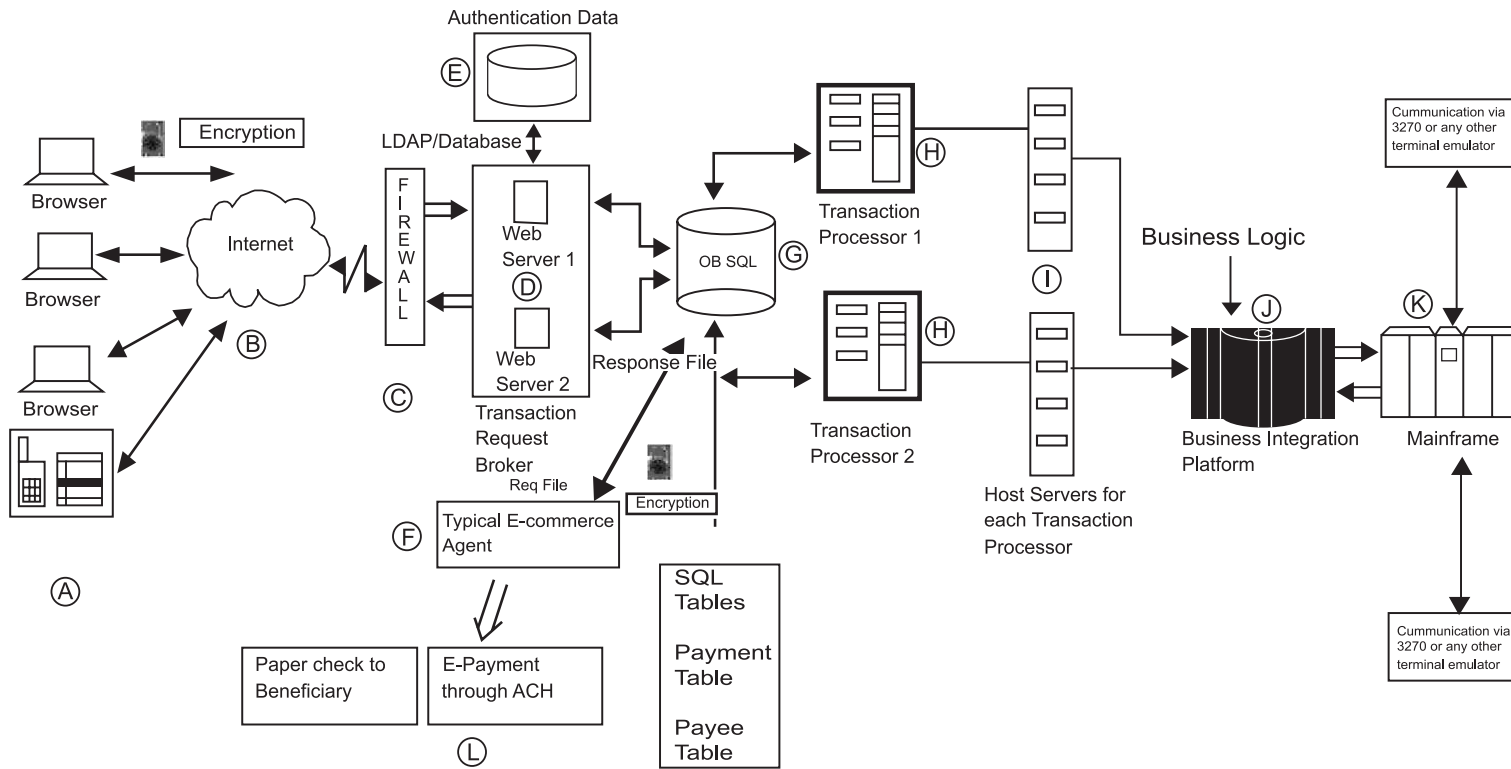


Fig. 2. System architecture of a typical electronic bill-pay system.

Table 1  
Legend for the architecture in Fig. 2

Component label	Component name
A	Browser/Mobile Client
B	Internet
C	Firewall
D	Web Server
E	LDAP/Database Server
F	E-Commerce Agent
G	Database
H	Transaction Processor
I	Host server for the transaction processor
J	Business Integration Platform
K	Mainframe
L	E-Payment recipient via ACH network <sup>a</sup>

<sup>a</sup> Out of our scope of analysis.

usually monitored by an intrusion detection system 24 h a day.

Inter-Financial Institution communication commonly occurs on SWIFT networks that use Interactive Financial Exchange (IFX) interface, which is an

improvement of the Open Financial Exchange (OFX) and GOLD specifications. Some of the other emerging technologies which bill pay service providers are incorporating are:

- *Electronic Data Interchange for Internet Applicability Specification (AS2)*, a draft security standard based on HTTP that ensures security and non-repudiation through encryption and digital signatures.
- *The HTTP-based Electronic Business XML (ebXML) messaging service.*
- *The open-network payment-card Secure Electronic Transaction™ Protocol* that ensures security of e-transactions over the Internet.

#### 2.4. Communication channels

Various communication channels can be used by the consumers to access the bill-payment application

Table 2  
Protocol/components and their functional description

Node labels	Protocol used	Component	Functional description
A–B	HTTPS/WAP	WAP/Browser–Internet	Desktop/Mobile Client uses to view his/her e-payment information
C–D	SSL/SET	Internet–Firewall	Uses secure HTTP using SSL (a synchronous communications protocol) to communicate with the Web server, or a scheduled transaction via SET (secure message format that may be used asynchronously)
D–E	LDAP/Database	Web server–LDAP/Database	The Web server uses LDAP/Database queries to authenticate the user
D–G	DCOM	Web server–Database	Distributed COM is the inter-process communications protocol used by the Web server when communicating with the DB component
G–F	FTP (with PGP)	Database–e-Commerce agent	The Database Server sends the bill payment to the e-commerce agent via FTP
G–H–I	SQL over Sockets	Database–Business Integration Servers	The Business Integration Platform and the Web server communicate with the SQL server using TCP/IP sockets
I–J	SQL over Sockets	Transaction Processor	Used for balancing load
J–K	TPS/3270	Mainframe–terminal	The back-end users the TPS/3270 protocol that is proprietary to query the mainframe and verify that the updates are regularly done
F–L	Proprietary ACH	e-Commerce Agent–Payment recipient	The e-commerce agent uses the proprietary ACH protocol to make electronic payments on behalf of the consumer to the biller (out of the scope of the EBPP system)

that is hosted by the financial institutions. Some of the commonly used channels are:

- (1) *Browser-based access from desktops*: Commonly, users tend to access these applications either from home or work through their desktops with the help of browsers like Internet Explorer and Netscape. They use secure HTTP or HTTPS to open a secure channel between the client and the Web server hosted by the bill-pay system.
- (2) *WAP-based access from mobile clients*: Some more sophisticated and mobile users tend to use their PDAs, cell-phones, etc., to access these clients in a comparatively less reliable and secure environment through WAP. However, nowadays, attempts are being made to make even these systems more and more secure than they previously were.
- (3) *Private financial networks*: Other financial institutions and networks like ACH networks that access the system come under this category. They commonly use X.25, SNA and TCP/IP as their regular communication channels.

### 3. The transaction workflow analysis—an information assurance focus

The management of business processes and transactions in banking systems is a regular activity. The sheer volume and the criticality of protecting customer data pose a challenge to the analysis of these systems. This is because, in addition to the traditional database support, they also need synchronization of work, cooperation between concurrent workflows and non-

serializable access to shared resources. Also, the transactions are vastly more voluminous than other traditional database applications [2].

A workflow can be defined as a composite set of tasks comprised of coordinated computer-based and human activities. According to Casati [4], in fact, a workflow can be defined as follows:

- An integration of software tools for automating and improving business processes.
- A process consisting of a number of individual tasks that need to be coordinated to achieve a particular business goal.

To capture the various nuances of this transaction from the threat focus, and to get a better understanding of the components involved, a workflow representation is developed [13]. We have attempted to analyze information assurance issues in EBPP systems through three activity-based workflows that function as process monitors as well as process analyzers for representing the EBPP system. Though there would be issues regarding work practices and choices in architecture of the EBPP system that will make the workflows differ from the ones we suggest below, we feel that the ones below could be used as blueprints that guide and shape those actions. Since workflows are abstract constructions that form a common reference model which assist in representing the external world [3], these workflows can be adapted. We will call ad hoc flow the possibility of altering the flow of a particular workcase. Because of its particular characteristics, a workcase may have to follow a different sequencing of activities from the one planned for more standard workcases of that type.

Table 3  
Typical EBPP workflows and their classification

Key area	Workflow	Transaction volume	Traits
Administration	The Bill-Pay Account-Index Setup Workflow (described in Section 3.1)	Moderate	Database-oriented, origination–authentication issues involved, Primary Users: Insiders/Administrators
Day-to-day customer transactions	The Bill-Payment Workflow (described in Section 3.2)	High	Day-to-day transactions and activities
The integrated reconciliation process	The end-to-end transaction in an EBPP system (described in Section 3.3)	Low	Overall payment system with consolidated daily systems conciliation

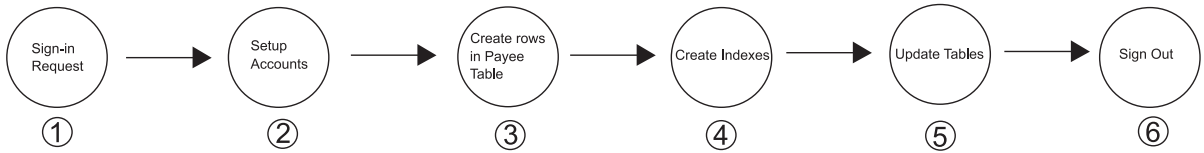


Fig. 3. Account-index setup system state diagram.

There are different forms of ad hoc flow discussed in the recent literature: Ad hoc planning is the case where a particular actor in the workflow may alter the plan of activities of the work case [8,10].

We have identified three basic workflows that can encapsulate most of our security assessment issues with an information assurance focus. A framework has then been developed to conduct a transactional analysis of the above workflows to bring forth those issues.

For complete coverage of all the possibilities of information assurance, we based our workflows on three key areas owing to their operational traits like transaction volumes, administration and day-end reconciliation business processes as summarized in Table 3.

The last workflow in Table 3 is a holistic one that captures an end-to-end functionality of the entire system. Superficially, it might appear as a combination of the two prior workflows. However, in addition to those activities it also captures the back-office business processes like FTP transmission to the e-commerce

agent, direct daily updates to the database via 3270 mainframe terminals based on the response sent by the e-commerce agent, etc., that are transparent to the EBPP system end-user. However, they open up different and newer fronts for raising serious information security concerns that must not be overlooked. Examples include capturing the file sent to the e-commerce agent when transmitted using FTP, internal threats posed to the EBPP’s mainframe system when operated by the employee responsible for it.

3.1. The bill-pay account-index setup workflow

This workflow, as introduced in Table 3, is an administrative transaction. The transaction volume for such a transaction is moderate. Furthermore, the number of customers that register online everyday is on the rise.

The following workflow/state diagram (Fig. 3) basically populates a database table that maps the

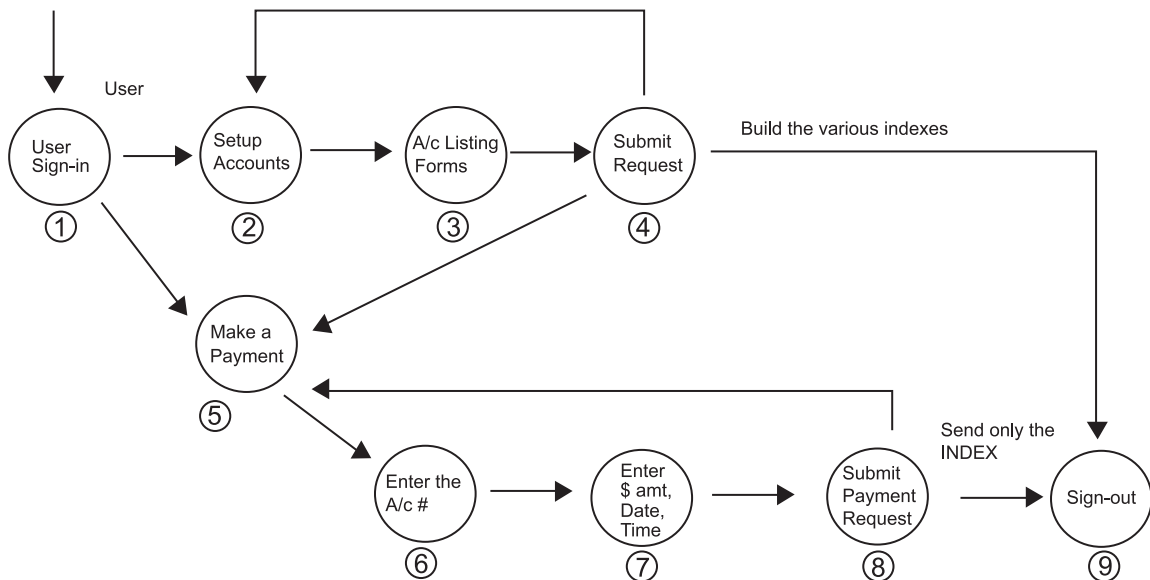


Fig. 4. Bill-payment state diagram.

customer's account number/routing number to a persistent index value, which is then used for future payment transactions that are made by that customer. This is a very safe technique of transmitting payment information. A detailed discussion on this issue can be found in Section 5.

### 3.2. The bill-payment workflow

This is the most frequently used transaction in the EBPP system. Here, the administrative workflow is used for setting up new accounts. Thus, this transaction is most likely to face information security breach attacks. Indexed account numbers are utilized here to execute daily payment transactions vis-à-vis the actual account number/routing number combinations. The workflow in Fig. 4 shows how the indexed account number is sent along with the other payment details when the payment request is submitted.

### 3.3. The end-to-end transaction in an EBPP system

The overall state diagram that illustrates all the states in an end-to-end transaction encompassing multiple players in the entire billing process is shown

in Fig. 5. The flow is a re-cap of the transactions already discussed. However, it also involves a file exchange with the e-commerce agent using FTP, which provides another opportunity for hackers to attack.

The next two sections introduce the Adapted STRIDE model for threat analysis and discuss the threats that the system in each of the above transactions is vulnerable to, in the light of that model.

## 4. The adapted STRIDE model [9]—the threat taxonomy

How can an attacker change the authentication data?

What's the impact if an attacker can read the system data?

What happens if access is denied to the system database?

To aid in asking these kinds of pointed questions, we argue for the use of threat categories. Therefore, we adapt and extend the STRIDE threat model. Developed by Microsoft, STRIDE is an acronym derived from the following six threat categories: Spoofing identity (S), Tampering with data (T),

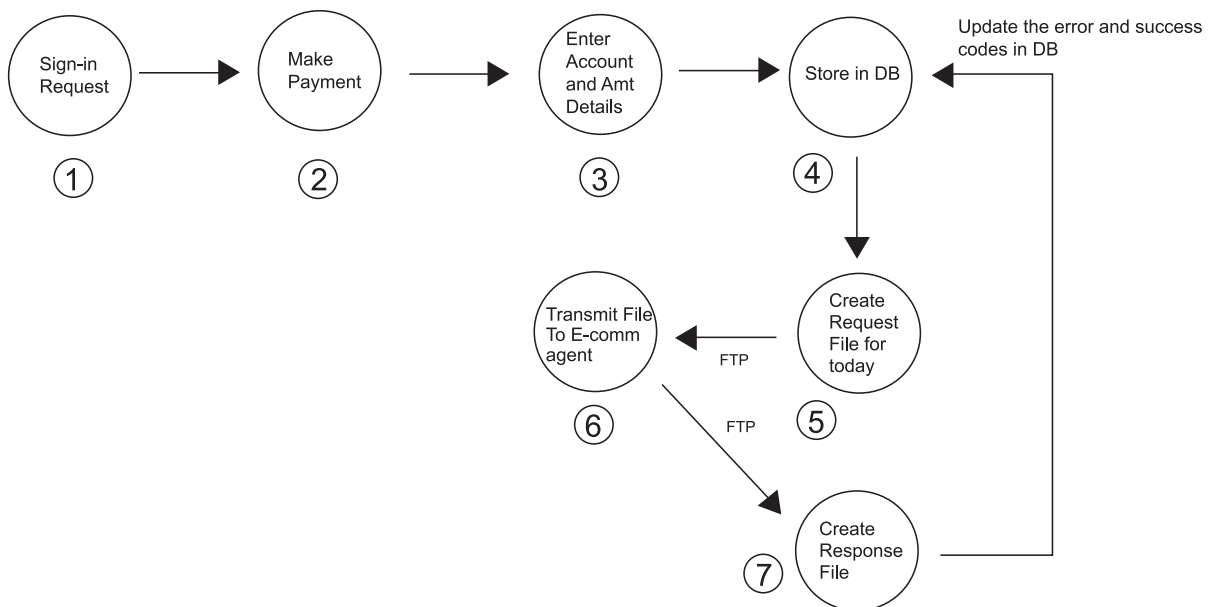


Fig. 5. The generic bill-payment system state-transition diagram.

Repudiation (R), Information disclosure (I), Denial of service (D) and Elevation of privilege (E) [9].

#### 4.1. Inter-relation of threats and other forms of attacks

In fact, the above threat categories may not be mutually exclusive. A threat that is exploited can lead to other threats. Some threat types can interrelate. It is not uncommon for information disclosure threats to lead to spoofing threats if the user's credentials are not secured. Of course, elevation of privilege threats are, by far, the worst threats—if someone can become an administrator or can get to the root on the target computer, every other threat category becomes a reality. Conversely, spoofing threats might lead to a situation where escalation is no longer needed for an attacker to achieve his goal. For example, using SMTP spoofing, an attacker could send an e-mail purporting to be from the CEO and instructing the workforce to take a day off.<sup>3</sup>

#### 4.2. Typical threats faced by EBPP systems

EBPP systems often have to face several kinds of attacks. Some of the threats include the following:

*Threat no. 1:* A malicious user views or tampers with the consumer's billing data en route from the Web server to the client or from the client to the Web server (Tampering with data/Information disclosure).

*Threat no. 2:* A malicious user views or tampers with the billing data en route from the Web server to the COM component or from the component to the Web server (Tampering with data/Information disclosure).

*Threat no. 3:* A malicious user accesses or tampers with the billing data directly in the database (Tampering with data/Information disclosure).

*Threat no. 4:* A malicious user views the LDAP authentication packets and learns how to reply to them so that he can act "on behalf of" the user (Spoofing identity/Information disclosure/Elevation of privilege [if authentication data is that of an administrator]).

*Threat no. 5:* A malicious user defaces the Web server by changing one or more Web pages (Tampering with data).

*Threat no. 6:* An attacker denies access to the bill-pay database server computer by flooding it with TCP/IP packets (DoS).

*Threat no. 7:* An attacker deletes or modifies the audit logs (Tampering with data/Repudiation).

*Threat no. 8:* An attacker places his own EBPP's Web server on the network after killing the real system server with a distributed DoS attack (Spoofing identity, in addition, a particularly malicious user could instigate all threat categories by stealing passwords or other authentication data, deleting data, etc.).

*Threat no. 9:* An attacker deletes or modifies the data in the File sent to and fro from the Database of the EBPP system to the e-commerce agent, during the FTP transmission (Spoofing Identity/Tampering with data/Repudiation. It may, however, be noted that this threat is specific to the EBPP system and is not defined in the STRIDE classification).

*Threat no. 10:* An insider (e.g., an employee) deletes or modifies the data (request/response file for batch processing) that is being updated to the mainframe while accessing it through the TPS/3270 terminal (Tampering with data).

Note that this is a highly abridged list. For such a huge system, there could be many more threats. However, the above do cover a broad range of common threats. The above threats along with their numbers will be used to identify them in further discussion. We shall analyze typical transactions that are commonly exe-

<sup>3</sup> Threat Modeling, from devx.com: The know-how behind application development. <http://www.devx.com/codemag/Article/10338/0/page/3>.

cuted on such systems from a threat analysis perspective using the model that was introduced above.

## 5. Threat identification and analysis of EBPP vis-à-vis STRIDE model

### 5.1. Threat exposure and analysis

Table 4 is a result of the application of the Adapted STRIDE Model to Fig. 3 already discussed. The state numbers labeled in Fig. 3 are used for referring the state in which the application's threat is analyzed in every row of the table.

Due to security reasons, only during the account setup process the entire encrypted account numbers are

Table 4  
STRIDE analysis of workflow in Fig. 3

State of the system during the transaction	Component/ protocols used	Threats faced	Threat category
1—Sign-in request	HTTPS/SSL, LDAP	1, 4	1—T <sup>a</sup> , I 4—S, I, E
2—Setup accounts	HTTPS/SSL, DCOM, SQL	1, 2, 5, 8	1—T, I 2—T, I 5—T 8—S and possibly all of T, R, I, D and E
3—Create rows in payee table	HTTPS/SSL, DCOM, SQL	1, 2, 5, 8	1—T, I 2—T, I 5—T 8—S and possibly all of T, R, I, D and E
4—Create index	HTTPS/SSL, DCOM, SQL	1, 2, 5, 8	1—T, I 2—T, I 5—T 8—S and possibly all of T, R, I, D and E
5—Update tables	HTTPS/SSL, DCOM, SQL	1, 2, 5, 8	1—T, I 2—T, I 5—T 8—S and possibly all of T, R, I, D and E
6—Sign out	HTTPS/SSL	1	1—T, I

<sup>a</sup> T represents the 'Tampering with data' threat category as introduced earlier in the STRIDE Model in Section 4. The other alphabets also represent threat categories as mentioned earlier.

Table 5  
STRIDE analysis of workflow in Fig. 4

State of the system during the transaction	Component/ protocols used	Threats faced	Threat category
1—Sign-in request	HTTPS/SSL, LDAP	1, 4	1—T, I 4—S, I, E
2b–4b	HTTPS/SSL, DCOM, SQL	1, 2, 5, 6, 8*	1—T, I 2—T, I 5—T 8—S and possibly all of T, R, I, D and E*
5—Make a payment	HTTPS/SSL, DCOM, SQL	1, 6	1—T, I 6—D
6—Enter the account number	HTTPS/SSL, DCOM, SQL	1, 6	1—T, I 6—D
7—Enter the US\$ amount, date and time	HTTPS/SSL, DCOM, SQL	1, 6	1—T, I 6—D
8—Submit payment request	HTTPS/SSL, DCOM, SQL	1, 6	1—T, I 6—D
9—Sign out	HTTP/SSL	1	1—T, I

transmitted. On all other occasions of bill payments, only the index of the account is sent and not the entire account number. These indices are created by using common hashing techniques. By using an alias like an index for the account number, a significant amount of security can be achieved. This is because the index is then again used by the receiving bill-pay system to map it back to the appropriate account number using the same hashing algorithm. The advantage is that such an index if captured over a transmission line will be useless to the hacker without the knowledge of the hashing algorithm.

The Adapted STRIDE Model is now applied to Fig. 4, and Table 5 illustrates the analyzed threats faced by the system in that transaction.

It is easy to see that a transaction that involves sending an entire batch of account numbers is significantly more vulnerable due to the increased number of threat categories that it might face. However, during the process of sending the hashed index that symbolizes the account number only, a reduced set of threats are faced. This presents a significant advantage over sending the entire account number.

The threat model is now applied to the overall end-to-end transaction to get a comprehensive picture of the total threats faced by the EBPP system.

Table 6  
STRIDE analysis of workflow in Fig. 5

State of the system during the transaction	Component/ protocols used	Threats faced	Threat category
1—Sign-in request	HTTPS/SSL, LDAP	1, 4	1—T, I 4—S, I, E
2—Make payment	HTTPS/SSL, DCOM, SQL	1, 6	1—T, I 6—D
3—Enter account number and amount details	HTTPS/SSL, DCOM, SQL	1, 6	1—T, I 6—D
4—Store in DB	HTTPS/SSL, DCOM, SQL	1, 3, 6	1—T, I 3—T, I 6—D
5—Create request file for today	System generated	10	10—T.
6—Send request file to the e-commerce agent	FTP	9	9—S, T, R
7—Send response file back to EBPP for updating its database	FTP	3, 7, 9, 10	3—T, I 7—T, R 9—S, T, R 10—T

Applying the Adapted STRIDE Model to Figs. 4 and 5, we get the results as shown in Table 5 and Table 6.

## 6. Framework for risk and vulnerability assessment and measurement of a security rating of an EBPP system

The primary focus of this section is to develop metrics based on security properties to aid different kinds of decision making, such as risk management, resource allocation, program planning or selection of specific products or services. The suggested model may serve security managers as a basis for trend analysis, or justify requests for additional resources. While various IS assessments of products and technologies are available, those assessments result in narrative descriptions and do not incorporate metrics. Some of the guidelines and resources that can be utilized for security assessment are The Trusted Computer System Evaluation Criteria (TCSEC), Department of Defense (DoD) and Common Criteria (CC). The CC has introduced the concept of a Protection Profile (PP).

For the security managers to assess and measure risk exposure and security posture of EBPP system,

we suggest a step-by-step procedure to derive an overall security rating for the EBPP system as follows, and also shown in Fig. 6.

The steps are as follows:

- Step I. Security and criticality index assessment of individual components of the EBPP technical architecture.
- Step II. Determination of ratings for the STRIDE threats vis-à-vis the EBPP system of the financial institution.
- Step III. Mapping of the transaction flow states for the three state-transition diagrams across components.
- Step IV. Assessment of STRIDE threats faced by various states of the state diagrams.
- Step V. Computation of the vulnerabilities and normalized threat sets for states of the state diagrams.
- Step VI. Consolidation of state vulnerabilities at state diagram level.
- Step VII. Derivation of the final vulnerability/security score for the EBPP system through vulnerabilities of state diagrams.

### 6.1. Step I

We propose a framework to assess components of the technical architecture and assign security ratings to components. The technical architecture may be different from components and transaction views, but any architecture with  $N$  components may be represented as  $ARCH = \{C_A, C_B, \dots, C_N\}$ .

The ratings are on a rank-order comparative and ordinal scale of 1–10 with 1 implying the least secure and 10 the most secure.

Table 7 illustrates a rank-order comparative ordinal security rating of any component,  $N=C_N$ . For example,  $C_A$  is rank-order comparative and ordinal rating for component A, as identified in Fig. 2. Similarly, ratings for all the components should be assigned based on the same ordinal and comparative scale of 1–10.  $\Psi_A$  is a criticality index for component A, which determines how critical the component is in the architecture and within the level of security wrappers, which may be used to protect the integrity of the asset. We suggest that criticality index is a concept similar to probability and assess-

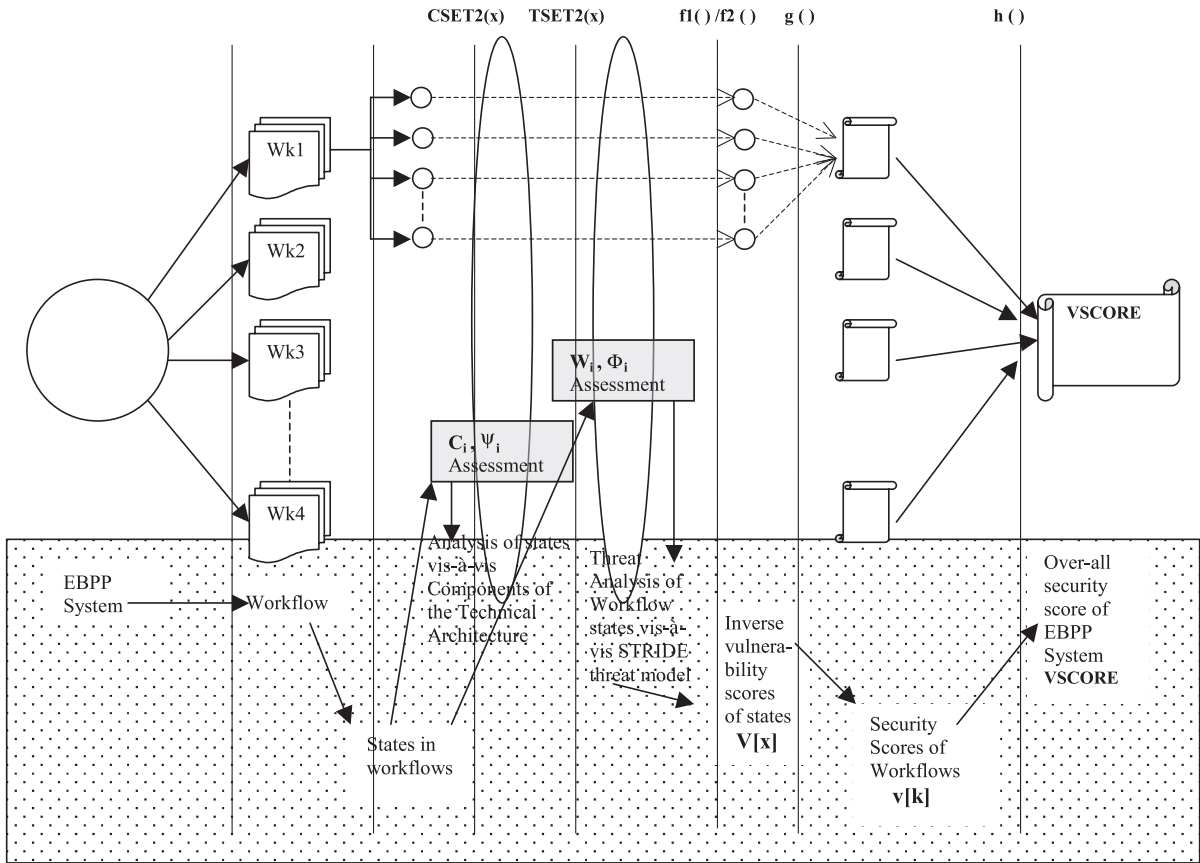


Fig. 6. The VSCORE Computation Process: the 7-step framework.

ment of it should be done with assignment of an index from 0 to 1.

Advisory Note: Security rating of any component reflects how “secure” that component is. Determination of such a rating can be based on various factors including patch management level of the component, physical security, experience and skills of the administrators, degree of complexity in maintenance, redundancy and interaction with other

components. The criticality index can be determined on the basis of importance of the component in any and all transactions that go through it. Some factors that can be used to assist in index value determination include fail-over options, transaction replay possibilities, redundancy, extent of damage that can be caused in case of any compromise, etc.

Note: Generally, the higher the adjudged criticality for any component, the higher the risk associated with it relative to other components, other factors kept constant. These are relative component security/vulnerability ratings and may well vary with the adopted policy with regard to risk assessment of the components in the individual firm. Even so, the methodology developed here would be useful if practiced systematically in an organization.

Table 7 Rank-order comparative ordinal security ratings of components

	Component											
	A	B	C	D	E	F	G	H	I	J	K	L
Security	$C_A$	$C_B$	$C_C$	$C_D$	$C_E$	$C_F$	$C_G$	$C_H$	$C_I$	$C_J$	$C_K$	$C_L$
Criticality index	$\Psi_A$	$\Psi_B$	$\Psi_C$	$\Psi_D$	$\Psi_E$	$\Psi_F$	$\Psi_G$	$\Psi_H$	$\Psi_I$	$\Psi_J$	$\Psi_K$	$\Psi_L$

### 6.2. Step II

The security manager, based on experience and judgment and consultation with concerned personnel, may determine comparative weights of impact of all the ten threats in case of exploitation.

$T_i$ = $i$ th threat, where  $\{0 < i < 10\}$ . The threats are as discussed in the earlier Section on STRIDE Model.

$W_i$ =weight of impact of threat  $T_i$ , where  $\{0 \leq i \leq 100\}$ .

We propose, to derive a measurement of vulnerability assessment of the EBPP system, the weights be assigned based on comparative ordinal rating on a scale of 1–100, where 1 is least impact and 100 the most.

$\Phi_i$ =probability of occurrence of the exploitation of  $i$ th threat, where  $\{0 \leq i \leq 1\}$ .

Advisory Note: STRIDE broadly classifies 10 kinds of threats that are possible. For any given implementation of EBPP system, the known as well as unknown vulnerabilities and exposures may create threats which will all not be equal. Some threats will be more damaging than others in any specific environment as will be their chances of being materialized. For instance, STRIDE threat 3 (tampering data directly in the database) is certainly more dangerous launching a [D] DOS attack a STRIDE threat type 6. The security manager can assess which threats are more real for their environment than others, based on their knowledge of their EBPP systems' configurations and architecture. An important note on difference between threat weights ( $W_i$ ) and component securities ( $C_i$ ) is that component security is security manager's assessment of "how secure a component is" and threat weight is "how real a STRIDE threat is to his EBPP environment".

### 6.3. Step III

The next step would be to chart out a transaction flow of each state of the state–transition diagrams of Section 3.

For example, state 1 of the account-index setup system state diagram would be a sign-in request and is represented in Fig. 7 as 3.1. The state 3.1 would have to traverse through components A, B, C, D and E as shown. In Fig. 7, a check-mark against any component for a state demonstrates that the particular

component is involved in the state against which it is marked.

We have, earlier, represented the set of all components as

$$\text{ARCH} = \{CA, CB, \dots, CN\}$$

where  $N$  is the number of components.

So, for each state, we will have a set of component security ratings that will be involved in the flow of the state. We have represented that state set of components as  $\text{CSET}(x)$  for state  $x$ .

$$\text{CSET}(x) \subseteq \text{ARCH}$$

For the purpose of calculation of the final VSCORE, the vulnerability/security score for the EBPP system, we have to develop an adjusted set of component security ratings, called  $\text{CSET2}(x)$  for any state  $x$ .

$$\text{CSET2}(x) = f_z(\text{CSET}(x))$$

Here, the  $f_z$  function is a simple product of the corresponding component security and critical index. This function could be any other mathematical operation depending upon the complexity of the dependence of components. So for each member of  $\text{CSET}(x)$ , we get a member for  $\text{CSET2}(x)$  by multiplying the corresponding security ratings and criticality indices.

For example, if  $C_i \in \text{CSET}(x)$ , then the corresponding member of  $\text{CSET2}(x)$  would be  $C_i \Psi_i$ , which is adjusted component security.

### 6.4. Step IV

Also from Table. 8, we get the threats that may be faced by this state during the execution, which are threats 1 and 4. The elliptical shapes contain the threats that the corresponding state of the workflow may encounter. The threats for any state  $x$  form a set  $\text{TSET}(x)$ . The possible values for this set  $\text{TSET}$ , for any state, will be a subset of  $\text{UTSET}$ .

$$\text{UTSET} = \{T1, T2, T3, T4, T5, T6, T7, T8, T9\}$$

$$\text{TSET}(x) \subseteq \text{UTSET} \text{ for any state of workflow } x.$$

Normalized  $\text{TSET}(x)$  or  $\text{TSET2}(x)$  would be obtained after taking into consideration the probabilities( $\Phi$ ) for individual threats from Table 8. The set  $\text{TSET2}(x)$  can be obtained by multiplying threat

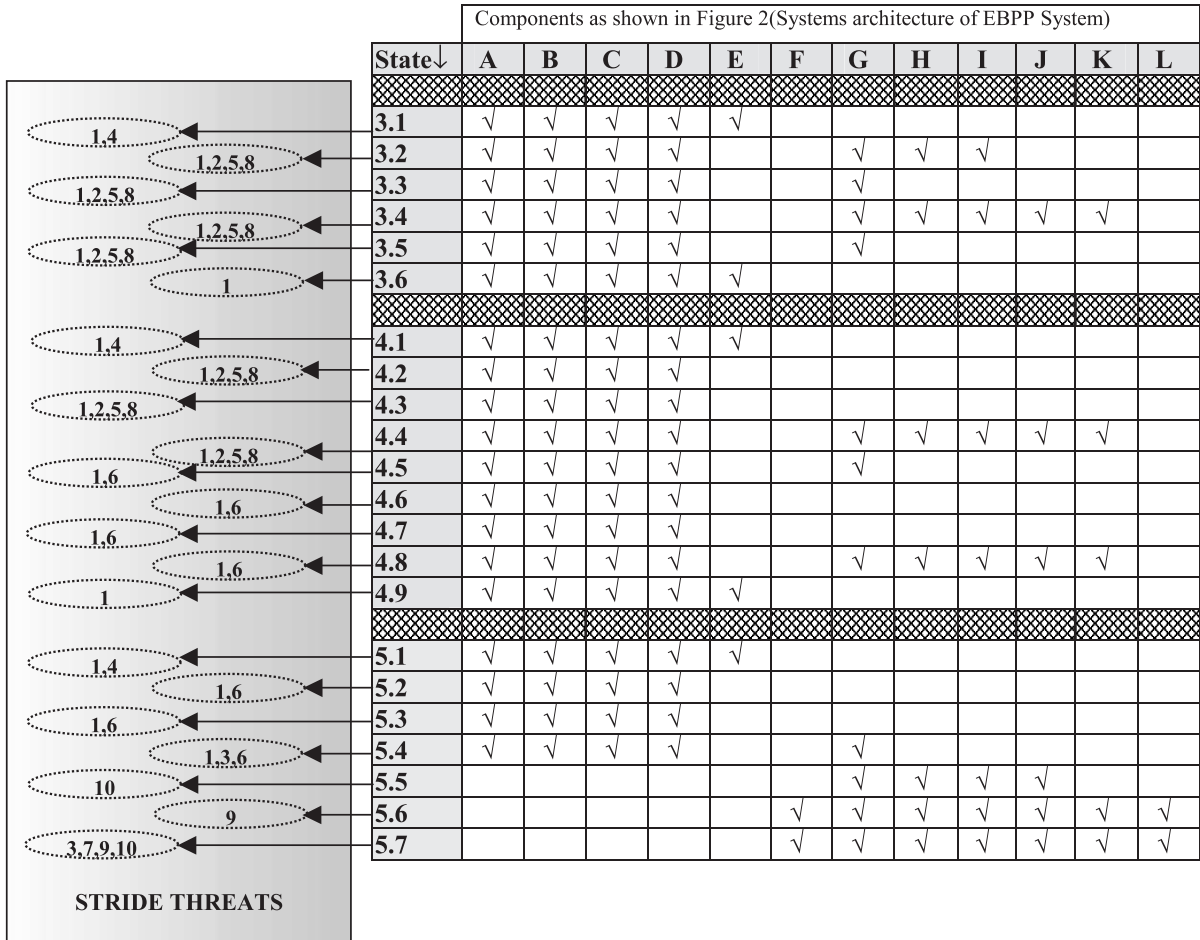


Fig. 7. The component-threat matrix used for VSCORE computation.

weight rating ( $W_i$ ) of member threat in TSET(x) with the corresponding probability  $\Phi_i$

$$TSET2(x) = f_y(TSET(x))$$

Here, we have the  $f_y$  function as a simple product of corresponding threat weight and probability.

So for each member of TSET(x), we get a member for TSET2(x) by taking the product of the corresponding weights and probabilities.

For example, if  $T_i \in TSET(x)$ , then the corresponding member of TSET2(x) would be  $W_i \Phi_i$ , which is normalized STRIDE threat.

### 6.5. Step V

We propose that vulnerability of any state  $V[x]$ , where  $x$  is the state's representative level/name, would be a function of set of security ratings of the individual components involved in state and the STRIDE threats identified for the state. For the technical architecture we presented in Fig. 2, the table created would be as shown in Fig. 7 for the EBPP system.

Table 8  
Comparative ordinal rating of threat impact and probabilities

	STRIDE threat number								
	1	2	3	4	5	6	7	8	9
Threat weight rating	$W_1$	$W_2$	$W_3$	$W_4$	$W_5$	$W_6$	$W_7$	$W_8$	$W_9$
Threat probability rating	$\Phi_1$	$\Phi_2$	$\Phi_3$	$\Phi_4$	$\Phi_5$	$\Phi_6$	$\Phi_7$	$\Phi_8$	$\Phi_9$

Let the set of component security ratings for any state be CSET2( $x$ ) and normalized STRIDE threat set be TSET2( $x$ ).

Thus,

$$V[x] = f_2(\{\text{TSET2}(x)\}) \text{ op1 } f_1(\{\text{CSET2}(x)\})$$

Note: The values for CSET2 are based on subjective judgments of reviewers.

For vulnerability of state 1 of the state diagram of Fig. 3, we will have

$$V[3.1] = f_2(\{\text{TSET2}(3.1)\}) \text{ op1 } f_1(\{\text{CSET2}(3.1)\})$$

Where  $\text{CSET2}(3.1) = \{c_A, c_B, c_C, c_D, c_E\}$ , and  $\text{TSET2}(3.1) = \{t_1, t_4\}$ .

Here,  $c$  and  $t$  are adjusted component security ratings and normalized STRIDE threats, respectively. For example,  $c_A = C_A * \Psi_A$ ,  $c_B = C_B * \Psi_B$ ,  $c_C = C_C * \Psi_C$ , and so on, and  $t_A = W_A \Phi_A$ , and so on. Also, function  $f_1$  is defined as average of members of CSET2( $x$ ) and  $f_2$  as average of members of TSET2( $x$ ).

Thus,  $f_1^n \rightarrow [\sum c_j]/n$ , where  $j$  and  $n$  depend on the set of  $c$ 's for that particular state (Fig. 7), and for only where  $c \neq 0$ .

Similarly,  $f_2^m \rightarrow [\sum t_i]/m$ , where  $i$  and  $m$  depend on the set of  $t$ 's for that particular state (Fig. 7), and for only where  $t \neq 0$ .

Finally, op1, is a division function<sup>4</sup> where the divisor is  $f_1\{\text{CSET2}(x)\}$  and the dividend is  $f_2\{\text{TSET2}(x)\}$ , where  $f_2\{\text{TSET2}(x)\} \neq 0$ . The quotient yields the vulnerability score (henceforth called the VSCORE). This quotient will always lie between 1 and 100. The higher score denotes a more vulnerable posture of that particular state of the concerned workflow.

<sup>4</sup> Discussion note: The decision of selection of operator op1 was primarily based on desired characteristics and objectives of our model. Our approach finally leads to an absolute metric, VSCORE, but serves primarily as a relative metric, allowing the user to compare the security (expressed through vulnerabilities in our model) of different versions of the same system, or to compare securities of different applications. Since most of the observations and subjective assessments would be from the managers and/or system administrators, we wanted to keep the model as simple as possible, without compromising on any of the characteristics for security measurement in relative terms.

## 6.6. Step VI

For each transition–state diagram, we can consolidate overall vulnerability score for the diagram. This can be achieved through a function of individual vulnerabilities of the state  $V[x]$ .

For any state diagram or transaction flow,  $v[k] = g(V[9], V[1], \dots, V[i])$ , where  $i$  is the number of states in the diagram for the  $k$ th workflow.

After computing  $V[x]$ s for all the states in a state–transition diagram, we can get an inverse vulnerability score for the whole workflow/state–transition diagram with simple average of all non-zero  $V[x]$ s.

So,  $g()$  is a simple average function of all non-zero state inverse vulnerability scores. i.e.,

$v[k] = (\sum V[i])/I$ , for  $I$  number of states with non-zero state inverse vulnerability scores, i.e., where  $V[i] \neq 0$ .

## 6.7. Step VII

The overall security score for the whole EBPP system is based on number of major state–transition diagrams and number of states in the diagram. So, it can be expressed as a function of vulnerability scores obtained for state diagrams in Step VI.

The security score of the whole system can be represented as :

$\text{VSCORE} = h(v[9], v[1], \dots, v[m])$ , where  $m$  is the number of state–transition diagrams.

For the matter of simplicity, we suggest that function  $h()$  again to be a simple average of non-zero security scores of state diagrams/workflows. (Note: We have presented one systematic method for computing a vulnerability score. Similar scores can be obtained for other situations as well for similar systems and an iterative policy-based feedback approach could be adapted for deriving the scores and comparisons of security strength over time or phases.).

## 7. VSCORE computation for the EBPP system in Fig. 2

Having developed and explained the framework for computing the vulnerability score for any technical architecture, we now demonstrate its calculation for

our example system whose architecture is shown in Fig. 2.<sup>5</sup> We now demonstrate the calculations step-wise as introduced in Section 6.

As explained earlier, the IT security managers must assess and measure risk exposure and the security posture of EBPP system. Also, based on experience and judgment and consultation with concerned personnel, one can determine comparative weights of impact of all the 10 threats in case of exploitation. Assuming that the IT security manager of our system in Fig. 2, with his own analysis and benchmark comparison, comes up with the following estimates of the ordinal ratings and the security ratings for the components mentioned.

### 7.1. Step I

Table 9 illustrates the various ordinal ratings assigned to the components of the technical architecture and the critical index values that are given to them.

The following constraints are observed:

- (1)  $0 \leq C_A \leq 10$  and
- (2)  $0 \leq \Psi_A \leq 1$

The security ordinal ratings are on a scale from 1 to 10 with 1 as the most secure and 10 as least secure. Also, the criticality indices for the components are assumed to be normalized and hence lie between 0 and 1. Thus, logically, the firewall is assumed to be the most secure with a rating of 9, the database being the most critical component is next best with a rating of 7, since it would be reasonably secure. The mainframe is also secure but is given a rating of 5 to account for internal threats due to the direct access through the 3270 terminals. The mobile/desktop client with its digital signatures and SSL certificates are assumed to be moderately secure with a rating of 6 and the Internet is given a minimum rating of 3 and so on.

Next, the criticality index has been assigned on the basis of how important the component is from the

Table 9

Security ratings and criticality indices of the components

Components	Security ordinal rating ( $C_A$ )	Criticality index ( $\Psi_A$ )	Adjusted component security rating ( $C_A \Psi_A$ )
A	6	0.6	3.6
B	3	0.2	0.6
C	9	0.8	7.2
D	7	0.7	4.9
E	8	0.8	6.4
F	5	0.6	3
G	7	0.8	5.6
H	4	0.5	2
I	4	0.5	2
J	6	0.7	4.2
K	5	0.7	3.5
L	5	0.6	3

point of its importance as regard to the business value, its value from the customer's requirements, the loss that the system and the organization would face, if it were to be compromised. Thus, the LDAP server, the Web server and the firewall are given a rating of 0.8. Similarly, the database, the mainframe and the business integration platform are next in the order of criticality with a rating of 0.7 and so on. The last column then computes the adjusted security rating as a product of the security rating ( $C_A$ ) and the criticality index ( $\Psi_A$ ) as ( $C_A \Psi_A$ ).

### 7.2. Step II

This step is similar to Step I in the sense that the threats that were earlier identified for the system need to be assigned a weight. This is again the IT security manager's assessment of the impact that the threat might have on the system if it is exploited and the system is under attack. Again, the ratings would be given in the range from 0 to 100. Like in the previous step, a relative judgment and assessment would be made for each of the threats. So the threats that attack the Web server, the COM component and the database are given a higher weight, say, 50. The others like the FTP transmission and DoS attacks, which although raise considerable concern, but may not have serious impacts have been assigned a relatively lower score of 40 and so on.

Next, the security personnel also need to input their estimates of the probabilities that these threats would

<sup>5</sup> Note that all numbers are hypothetical and might vary depending upon the IT security personnel's perception, judgment and hence assessment of the security measures that are in place for the system under consideration.

transform into actual attacks that compromise the system. Again, based on our judgment and intelligent guesses, we have enlisted the following ratings and probabilities for the threats in Table 10. However, we would again like to emphasize the fact that these numbers would be very individualistic and greatly depend on the nature of the organization, the security policies in place within it, and the security manager’s assessment of the threats and other related parameters.

Again, the following constraints are observed:

$$(0 \leq W_i \leq 100) \text{ and } (0 \leq \varphi_i \leq 1)$$

where  $W_i$  = the weight associated with the  $i$ th threat as explained in the previous section and  $\varphi_i$  = probability that the  $i$ th threat will be exploited and the system would be attacked.

Finally, the last column called the normalized threats is computed as a product ( $W_i \varphi_i$ ) of the weights and the probabilities of these threats.

### 7.3. Step III

The first two steps basically list the assumptions and then assign proper numerical values to various parameters. In this step, Fig. 7 is used to create the sets of the components that are involved in each of the states. These sets are known as CSET( $x$ ) where  $x$  = state of the transaction and the set contains the components involved to execute that state.

Having created the CSET( $x$ ), it is now easy to create the CSET2( $x$ ) for the corresponding state in the transaction. Basically, the components need to be

Table 10  
Threat ratings and their weights along with their normalized values

Threats	Threat weight rating ( $W_i$ )	Threat probability ( $\varphi_i$ )	Normalized threats ( $W_i \varphi_i$ )
1	50	0.6	30
2	50	0.8	40
3	50	0.3	6
4	20	0.9	36
5	40	0.6	18
6	30	0.6	18
7	30	0.3	12
8	40	0.5	20
9	40	0.9	22.5
10	25	0.8	20

replaced with their adjusted security ratings from step 1, and the CSET2( $x$ ) for that particular state of the state diagram would be created. Quite understandably, the rating would be zero for that component that does not feature in the execution of that state.

Thus, e.g., let  $x=3.1$ , i.e., the state 1 for Fig. 3. The CSET(3.1) = {A, B, C, D, E} or if the security ratings are used, then the CSET(3.1) = {6, 3, 9, 7, 8}.

Now, if the CSET2(3.1) is to be computed, we simply substitute the security ratings with the adjusted security ratings. Thus, CSET2(3.1) = {3.6, 0.6, 7.2, 4.9, 6.4}. This can similarly be done for the remaining states and for all the state diagrams as well.

### 7.4. Step IV

Similarly, the threat sets TSET( $x$ ) and TSET2( $x$ ) now need to be computed for each of the states of the state diagrams. However, in making the TSET( $x$ ), one must use the pure threat numbers 1–10 that are already identified in Section 4. Thus, continuing our example for the state 4.1, TSET( $x$ ) can be computed as:

$$\begin{aligned} \text{TSET}(3.1) &= \{1, 4\}, \text{ and hence, } \text{TSET}(3.2) \\ &= \{1, 2, 5, 8\} \end{aligned}$$

and hence, TSET(3.2) = {1, 2, 5, 8} and so on, as shown in Fig. 7.

Now, the next step as in the previous step is to arrive at the TSET2 values. Again, quite symmetrically, we need to just substitute the normalized threat values ( $W_i \varphi_i$ ) instead of the pure threat numbers for each of the threats that appear in TSET for that state. Thus,

$$\begin{aligned} \text{TSET2}(3.1) &= \{30, 36\}, \text{ and hence, } \text{TSET2}(3.2) \\ &= \{30, 40, 18, 20\} \end{aligned}$$

Thus, similarly, the remaining TSET2 values can be computed for the remaining states and for all other state diagrams as well.

### 7.5. Step V

Next, we need to compute the vulnerability  $V(x)$  for each state. This is done in three steps. We first, compute  $f_1\{\text{CSET}(x)\}$  by taking the non-zero average

Table 11  
Computation of  $f_1\{CSET2(x)\}$

State↓	A	B	C	D	E	F	G	H	I	J	K	L	$f_1\{CSET2(x)\}$
3.1	3.6	0.6	7.2	4.9	6.4	0	0	0	0	0	0	0	4.540
3.2	3.6	0.6	7.2	4.9	0	0	5.6	2	2	0	0	0	3.700
3.3	3.6	0.6	7.2	4.9	0	0	5.6	0	0	0	0	0	4.380
3.4	3.6	0.6	7.2	4.9	0	0	5.6	2	2	4.2	3.5	0	3.733
3.5	3.6	0.6	7.2	4.9	0	0	5.6	0	0	0	0	0	4.380
3.6	3.6	0.6	7.2	4.9	6.4	0	0	0	0	0	0	0	4.540
4.1	3.6	0.6	7.2	4.9	6.4	0	0	0	0	0	0	0	4.540
4.2	3.6	0.6	7.2	4.9	0	0	0	0	0	0	0	0	4.075
4.3	3.6	0.6	7.2	4.9	0	0	0	0	0	0	0	0	4.075
4.4	3.6	0.6	7.2	4.9	0	0	5.6	2	2	4.2	3.5	0	3.733
4.5	3.6	0.6	7.2	4.9	0	0	5.6	0	0	0	0	0	4.380
4.6	3.6	0.6	7.2	4.9	0	0	0	0	0	0	0	0	4.075
4.7	3.6	0.6	7.2	4.9	0	0	0	0	0	0	0	0	4.075
4.8	3.6	0.6	7.2	4.9	0	0	5.6	2	2	4.2	3.5	0	3.733
4.9	3.6	0.6	7.2	4.9	6.4	0	0	0	0	0	0	0	4.540
5.1	3.6	0.6	7.2	4.9	6.4	0	0	0	0	0	0	0	4.540
5.2	3.6	0.6	7.2	4.9	0	0	0	0	0	0	0	0	4.075
5.3	3.6	0.6	7.2	4.9	0	0	0	0	0	0	0	0	4.075
5.4	3.6	0.6	7.2	4.9	0	0	5.6	0	0	0	0	0	4.380
5.5	0	0	0	0	0	0	5.6	2	2	4.2	0	0	3.450
5.6	0	0	0	0	0	3	5.6	2	2	4.2	3.5	3	3.329
5.7	0	0	0	0	0	3	5.6	2	2	4.2	3.5	3	3.329

Table 12  
 $f_2\{CSET2(x)\}$  computation values for the various states

State↓	1	2	3	4	5	6	7	8	9	10	$f_2\{TSET2(x)\}$
3.1	30	0	0	36	0	0	0	0	0	0	33
3.2	30	40	0	0	18	0	0	20	0	0	27
3.3	30	40	0	0	18	0	0	20	0	0	27
3.4	30	40	0	0	18	0	0	20	0	0	27
3.5	30	40	0	0	18	0	0	20	0	0	27
3.6	30	0	0	0	0	0	0	0	0	0	30
4.1	30	0	0	36	0	0	0	0	0	0	33
4.2	30	40	0	0	18	0	0	20	0	0	27
4.3	30	40	0	0	18	0	0	20	0	0	27
4.4	30	40	0	0	18	0	0	20	0	0	27
4.5	30	0	0	0	0	18	0	0	0	0	24
4.6	30	0	0	0	0	18	0	0	0	0	24
4.7	30	0	0	0	0	18	0	0	0	0	24
4.8	30	0	0	0	0	18	0	0	0	0	24
4.9	30	0	0	0	0	0	0	0	0	0	30
5.1	30	0	0	36	0	0	0	0	0	0	33
5.2	30	0	0	0	0	36	0	0	0	0	33
5.3	30	0	0	0	0	36	0	0	0	0	33
5.4	30	0	6	0	0	36	0	0	0	0	24
5.5	0	0	0	0	0	0	0	0	0	36	36
5.6	0	0	0	0	0	0	0	0	36	0	36
5.7	0	0	6	0	0	0	36	0	36	36	28.5

of the adjusted security ratings that appear in the set CSET2(x) for the state x. For example,

$$f_1\{CSET2(3.1)\} = \{3.6 + 0.6 + 7.2 + 4.9 + 6.4\}/5 = 4.54.$$

Next, we calculate  $f_2\{TSET2(x)\}$ , which again is a non-zero average of the normalized threats that appear in the corresponding set TSET2(x). Thus, for state 3.1, we get

$$f_2\{TSET2(3.1)\} = \{30 + 36\}/2 = 33.$$

Table 11 shows the values of  $f_1\{CSET(x)\}$ . Table 12 shows the computation of  $f_2\{TSET2(x)\}$ . Now,  $V(x)$  is defined as:

$$V(x) = \frac{f_2\{TSET2(3.1)\}}{f_1\{CSET2(3.1)\}}.$$

Thus,  $V(3.1)=33/4.54=7.27$ .

Similarly, the vulnerability of each of the other states can be computed. Table 13 illustrates the vulnerabilities for each state.

### 7.6. Step VI

We, now compute the vulnerability for a transaction as a whole. For this, we just consolidate the individual vulnerability scores for each of the states using an aggregation function like NON-ZERO AVERAGE( $x_1, x_2, x_3, x_4, \dots$ ), i.e.,  $v[k]=g(V[k_1], V[k_2], V[k_3], \dots)$ , where  $k$ =a particular workflow and  $k_1, k_2, k_3$ , etc., are the individual states for that workflow and  $g(\ )=(\sum V[k_i]/I)$  for  $i$  number of states in the transaction workflow. We thus arrive at the following

Table 13  
Vulnerability  $V(x)$  for each state

State↓	$V[x]$	State↓	$V[x]$
3.1	7.27	4.6	5.89
3.2	7.30	4.7	5.89
3.4	6.16	4.8	6.43
3.5	7.23	4.9	6.61
3.6	6.16	5.1	7.27
		5.2	8.10
4.1	7.27	5.3	8.10
4.2	6.63	5.4	5.48
4.3	6.63	5.5	10.43
4.4	7.23	5.6	10.82
4.5	5.48	5.7	8.56

Table 14  
Vulnerability scores for each state diagram

Transaction	$V[n]$
3	6.83
4	6.45
5	8.39

vulnerability scores (Table 14) for the workflows shown in Figs. 4, 5 and 6.

### 7.7. Step VII

Finally, we now compute the overall vulnerability score for the entire architecture presented in Fig. 2. This is again consolidated by the average function  $h(\ )$ , where  $h(\ )=(\sum v[j]/J)$ , where  $J$ =total number of transaction workflows present for the system. Thus,  $VSCORE=h(v[9], v[1], v[13], \dots)$ . For the system in Fig. 2, the overall VSCORE is shown in Table 15.

## 8. Conclusion

This paper has shown that the workflows depicting typical transactions that occur in EBPP systems on a regular basis are needed to develop threat metrics in the EBPP arena. Using the STRIDE Model, we identify the threats that such systems could potentially face. These threats along with the components of the architecture are then tied together in a framework model that may aid IT security managers of such systems to measure the security levels that are in place to protect them and monitor them regularly on a proactive basis. Finally, for the example system discussed in the paper, the step-wise computation of the VSCORE is also shown. The fact that the VSCORE encompasses and quantifies all the possible threats with its impact value uses sensible and realistic probabilities of attacks that exploit those threats, making it a useful metric. Moreover, the ability to attach weights to each threat makes it applicable to more generic systems beyond bill-payment systems

Table 15  
Overall vulnerability score for the system

System	VSCORE (s)
EBPP	7.22

per se. Further, this paper presents some research opportunities and areas for further in-depth analysis. A note of caution: A single score does allow risk-based assessments of the whole system and can serve as first step in hardening the system as well as the environment, yet, could appear deceptive and should not be taken as the final measure. The conceptual framework developed in this paper could be extended for like systems and a security benchmark analysis could be a definite outcome.

## 9. Further reading

[12]

## Acknowledgments

The authors would like to thank the anonymous referees for their critical comments and the editor-in-chief for his encouragement. The research of the third and fourth authors was funded by NSF under grant 0417095 and grant 0420448. The usual disclaimer applies.

## References

- [1] Alexandria Andreeff, Lisa C. Binmoeller, Eve M. Boboch, et al., E-Presentation and Payment. Is it just a click away? *Electronic Payments Journal*, Mar/April 2002, <http://www.nacha.org/newsletter/marchapril2002.pdf>, retrieved on April 5, 2003.
- [2] Anthony J. Bonner, Workflows, *Transactions and Datalog* (1999)<ftp://ftp.cs.toronto.edu/pub/bonner/papers/transaction.logic/pods99.ps> retrieved on June 27, 2003.
- [3] D.P. Bogia, S.M. Kaplan, Flexibility and control for dynamic workflows in the world's environment, in: N. Comstock, C.A. Ellis (Eds.), *Proceedings of the 1995 ACM Conference on Organizational Computing Systems (COOCS'95)*, Milpitas, California, ACM Press, New York, NY, USA, 1995, pp. 148–159.
- [4] F. Casati, S. Ceri, B. Pernici, and G. Pozzi, FSBG95: Conceptual Modeling of Workflows.
- [5] Gowrisankaran, Gautam, and Joanna Stavins, forthcoming, *Network Externalities and Technology Adoption: Lessons from Electronic Payments*, <http://www.econ.umn.edu/~gautam>, retrieved on 4/3/2003.
- [6] IBM Global Services, *Electronic Bill Presentment and Payment Systems—A Strategic Advantage*, <http://www-1.ibm.com/services/files/gss1247f.pdf>, retrieved on May 7, 2003.
- [7] Jeeto Patel, Christine Klima, *Technology–E-Payment Growth—Payers Hold Court*, Thomson Media, 2003 (June 23).
- [8] K.D. Swenson, R.J. Maxwell, T. Matsumoto, B. Saghari, K. Irwin, *A business process environment supporting collaborative planning*, CSCW'94, *Collaborative Computing*, vol. 1, Chapman & Hall, London, 1994, pp. 15–34.
- [9] Michael Howard, David LeBlanc, “The STRIDE Threat Model. From the Book ‘Writing Secure Code’”, Microsoft Press. Chapter 2: Designing Secure Systems, Pg. 38–60, 2002 Edition.
- [10] R. Blumenthal, G.J. Nutt, Supporting unstructured workflow activities in the Bramble ICN system, in: N. Comstock, C.A. Ellis (Eds.), *Proceedings of the 1995 ACM Conference on Organizational Computing Systems (COOCS'95)*, Milpitas, California, ACM Press, New York, NY, USA, 1995, pp. 130–137.
- [11] Silverline, Oasis, Diversinet and Intel. *Solution Blueprint: Secure, End-to-End Electronic Billing and Payment Solution*, <http://www.intel.com/ebusiness/pdf/affiliates/solutions/silverline0228.pdf>, retrieved on June 20, 2003.
- [12] The Interactive Financial Exchange Forum, IFX for EBPP Datasheet, <http://www.ifxforum.org/ifxforum.org/standards/ebpp.cfm>, retrieved on 4/3/2003.
- [13] Y.I. Song, H.R. Rao, and S. Upadhyaya, *Information Assurance Issues of the Workflow Management Systems in E-Banking: An Investigation on the Modal Points with High Risk*, University at Buffalo, Working paper, June 2003.