INTERNET EDI RISKS IN AN ORDERING SYSTEM

by

KETANSINH ANIRUDDHSINH GOHIL

SHORT DISSERTATION

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE

MASTER OF COMMERCE

IN

COMPUTER AUDITING

IN THE

FACULTY OF ECONOMIC AND MANAGEMENT SCIENCES

AT THE

RAND AFRIKAANS UNIVERSITY

STUDY LEADER: PROF. A. DU TOIT

JOHANNESBURG

OCTOBER 1997

DECLARATION

I declare that this short dissertation hereby submitted to the Rand Afrikaans University for the degree of Master of Commerce, except to the extent acknowledged in the text, is my own unaided work and has not been submitted previously for any degree to any other university.

KETANSINH ANIRUDDHSINH GOHIL

TABLE OF CONTENTS

INTERNET EDU RISIKO'S IN 'n BESTELSTELSEL

deur

KETANSINH ANIRUDDHSINH GOHIL

OPSOMMING VAN SKRIPSIE

INGEDIEN VIR DIE GRAAD MAGISTER COMMERCII IN REKENAAROUDITERING

IN DIE FAKULTEIT EKONOMIESE - EN BESTUURSWETENSKAPPE

AAN DIE RANDSE AFRIKAANSE UNIVERSITEIT

STUDIELEIER: PROF. A. DU TOIT

JOHANNESBURG

OKTOBER 1997

Opsomming

The Gartner Groep (1996c:1) het geruime tyd gelede reeds erkenning verleen aan die feit dat die Internet sedert sy stigting aansienlik uitgebrei het. Deesdae maak maatskappye seker dat hul toepassings by die Internet aanpas (The Gartner Group, 1997a:1), terwyl hulle die Internet ook ál meer inspan om hul transaksies te inisieer (The Gartner Group, 1996b:16). Hierdie tendens het reeds in Suid-Afrika ook gemanifesteer, soos byvoorbeeld by die Suid-Afrikaanse Lugdiens, wat 'n stelsel ontwerp het ingevolge waarvan vliegtuigkaartjies oor die Internet opgeveil kan word. So ook het Computicket 'n kaartjiebesprekingstelsel oor die Internet ingestel.

Die Suid-Afrikaanse Instituut van Geoktrooieerde Rekenmeesters vereis egter dat 'n ouditeur die kontroleveld vir elke wesenlike transaksiekategorie moet kan evalueer (SAIGR, SARG 400:18). Waar 'n organisasie dus 'n Internet-besteldiens ingestel het en sodanige stelsel die bron van wesenlike transaksies verteenwoordig, word daar van die ouditeur vereis om die kontroleveld van sodanige stelsel te kan evalueer. Watne en Turne (1994:14) het in hul navorsing daarop gewys dat, vanweë die immergroter gesofistikeerdheid van gerekenariseerde stelsels, dit nie meer prakties is vir die ouditeur om sy of haar oudit bloot *rondom* die rekenaar uit te voer nie; hy of sy sal voortaan sy of haar oudit *via* rekenaarstelsels moet uitvoer.

'n Literatuurstudie is onderneem ten einde ondersoek in te stel na navorsingsrisiko-modelle wat ontwikkel is om risiko's in Internet-besteldienste te identifiseer. Op grond van gemelde literatuurstudie is daar egter bevind dat daar nog nooit enige risikomodel ontwikkel is aan die hand waarvan 'n rekenaarouditeur sy of haar oudit kon uitvoer nie. Die hoofdoel van die onderhawige studie was gevolglik om die risiko's te identifiseer wat in Internet-besteldienste verskuil lê en om aan die hand van die geïdentifiseerde risiko's 'n risikomodel te ontwikkel wat ouditeurs sal kan inspan om die risiko's van Internet-besteldienste te evalueer.

Die navorsingsbenadering wat gevolg is, het die identifikasie behels van daardie tegnologiese komponente wat op Internet-besteldienste van toepassing is. Hierdie komponente is ondersoek en in besonderhede beskryf. Internet-besteldienste vereis oor die algemeen dat 'n bestelvorm (of 'n elektroniese vorm), soos in HTML (Ragget, 1997) omskryf, gebruik word. Hierdie bestelvorms word deur middel van HTTP op die kliënt-snuffelaar ("client browser") vertoon (Fielding, Irvine, Gettys, Mogul, DEC, Frystyk, Berners-Lee & MIT/LCS, 1997:36). HTTP word ook gebruik om die elektroniese bestelvorm aan die verskaffer se Web-bediener te kommunikeer (The Gartner Group, 1996a:35), wat die netwerkkommunikasieprotokol vir die Internet is (The Gartner Group, 1996b:45). Sodra die elektroniese bestelvorm ingevul en aan die kliënt se kant voorgelê is, word dit deur die verskaffer se Web-bediener ontvang, waarna 'n "Common Gateway Interface" ('n "CGI") gebruik word om 'n koppelvlak tussen die verskaffer se Internet-bediener en die interne bestellingstoepassingstelsel te voorsien (NCSA, University of Illinois by Urbana Champaign, s.j. a). Nadat hierdie komponente geïdentifiseer is, is hul werking in besonderhede bespreek.

Die volgende fase van die navorsing het die omskrywing van die begrip "risiko" behels, waarna 'n risikomodel geselekteer is wat sou aanpas by die omgewing wat geïnspekteer is. Die begrip "risiko" is hierna in terme van die kontroledoelwit omskryf. Watne en Turne (1990:308-309) se kontroledoelwitte is vir die doeleindes van hierdie skripsie geselekteer. Nadat die begrip "risiko" behoorlik gedefinieer is, moes 'n toepaslike risiko- of kontroledoelwitmodel vir toepassing in hierdie omgewing gekies word. Die twee modelle wat deur Boshoff (1985, 1990) ontwikkel is, te wete die "Access Model" en die "Path Context Model", is met die oog op die identifikasie van risiko's in Internet-bestelstelsels geselekteer. Ingevolge hierdie twee modelle moet die risiko in elke komponent van die toegangsroete ("access path") ontleed word.

HTML word hoofsaaklik gebruik om die elektroniese vorm te omskryf waarop die kliënt se bestelling geplaas word (Ragget, 1997). Die elektroniese vorm is in verskeie toevoerformate ("input formats") verkrygbaar, welke formate gewoneteks- en multi-lyninskrywings, kieslyste ("option menus") en rollyste

("scrolled lists"), radioknoppies en kontrolehokkies ("check boxes"), wagwoordinskrywings ("password entries") en terugstel- ("reset buttons") en invoerknoppies (submit buttons") insluit. The Gartner Group (1997a:1) erken egter dat die HTML e-vormtegnologie nie oor die nodige logika beskik om berekeninge en foutkontrole aan die kliënt se kant te doen nie, en aan die bediener se kant kompenseer dit nie genoeg vir berekeninge en foutkontrole nie. HTML-komponente voldoen gevolglik nie aan kontrolevereistes nie. Bepaalde uitsonderings is egter aangetoon.

Twee tipes HTTP's is geïdentifiseer, naamlik versoekboodskappe ("request messages"), wat van die kliënt na die verskaffer se bediener gestuur is, en responsboodskappe (response messages"), wat weer deur die verskaffer se bediener aan die kliënt versend is (Fielding *et al.*, 1997:11). Daar is bevind dat HTTP-boodskappe oor opskrifvelde ("header fields") beskik wat positief tot die gedefinieerde kontroledoelwitte bygedra het (Fielding *et al.*, 1997:97-139). HTTP-responsboodskappe stuur ook statuskodes terug wat 'n uitwerking op risiko het (Fielding *et al.*, 1997:53-65).

TCP/IP is ook bestudeer. Daar is bevind dat TCP oor opskrifvelde beskik, met bepaalde parameters wat 'n impak op risiko het. In die besonder is bevind dat TCP-pakkette in hul opskrifte oor kontrolesomme en bron-/bestemmingadresvelde beskik (Santifaller, 1994:32). Daar is voorts bevind dat gemelde velde 'n regstreekse invloed uitoefen op die akkuraatheid en geldigheid van data wat versend is. Daar is ook bevind dat IP oor opskrifvelde beskik (Santifaller, 1994:20-23). Op een uitsondering na is daar bevind dat hierdie opskrifvelde almal slegs op die IP-pakket betrekking het, en nie op die data wat daarin gestoor is nie (Santifaller, 1994:20-23). Hierbenewens beskik IP nie oor funksies vir einde-tot-einde-boodskapbetroubaarheid of vir vloeibeheer nie (Santifaller, 1994:19).

Die finale stap in die bestelproses is die voorsiening van 'n koppelvlak tussen die verskaffer se Internet-bediener en die internebestelling-stelsel (NCSA, University of Illinois by Urbana Champaign, s.j. a). CGI-tekste vertaal die HTML-elektroniese bestelvorm in 'n formaat wat deur die bestelstelsel aanvaar word

(NCSA, University of Illinois by Urbana Champaign, s.j. a). Die navorsing het nie 'n spesifieke implementering van CGI-tekste geselekteer nie, maar het eerder kwessies geëvalueer wat algemeen toepaslik op alle CGI-tekste is. Hierdie kwessies het spesifiek verband gehou met toepassingstelselomgewing-veranderlikes wat tydens die prosessering van die elektroniese bestelvorm na die CGI-tekste oorgedra is.

Tydens die proses ingevolge waarvan daar navorsing gedoen is oor 'n model vir die evaluering van die moontlike risiko's wat in 'n Internet-bestelstelsel skuilhou, is bepaalde bykomende areas vir navorsing blootgelê. In die besonder is drie verskillende implementerings van Elektroniese Data Uitruiling (EDU: "*EDI*") Internet-bestelstelsels geïdentifiseer, hoewel slegs een stelsel vir navorsing uitgesonder is. Die ander twee implementerings, wat die gebruik van die Internet as 'n koerier vir EDI-boodskappe en die vertaling of oorsit van Internet-bestellings in EDI-klagteboodskappe aan die verskaffer se kant behels (The Gartner Group, 1996b:4), mag moontlik potensiële areas vir navorsing oor risiko-evaluering oplewer. 'n Ander area wat vir navorsing blootgelê is, is die evaluering van risiko's in stelselspesifieke CGI-tekste.

Ten slotte kan daar sonder vrees vir teëspraak gestel word dat die uitbreiding van die elektroniese handel op die Internet tans deur twee kwessies gekortwiek word (The Gartner Group, 1997b:1), naamlik

- die kwelvraag of produkte wat via die Internet aangekoop is, belas moet word al dan nie;
- die kwessie of die uitvoer van gesofistikeerde enkripsie-sagteware vanaf die Verenigde State beperk behoort te word al dan nie.

Die suksesvolle oplossing van voormelde vraagstukke sal ongekende groei in Internet-handel ontketen (The Gartner Group, 1997b:1).

Synopsis

The Gartner Group (1996c:1) has recognised that the Internet has grown substantially since its inception. Recently, companies are "Internet enabling" their applications (The Gartner Group, 1997:1) and they are increasingly using the Internet to initiate transactions (The Gartner Group, 1996b:16).

The South African Institute of Chartered Accountants requires that an auditor should assess the control environment for each material class of transactions (SAICA, SAAS 400 .18). Therefore the auditor may be required to review an Internet ordering system where this is the source of material sales transactions. With the increasing sophistication of computerised systems, computer auditors are being forced to treat computer systems as the target of their audits (Watne and Turne, 1994:14). The literature survey conducted did not reveal a suitable risk model which may be applied to Internet ordering systems. Therefore the objective of the research was to develop such a risk model.

The research identified the technology involved in Internet ordering systems. A detailed literature review of these components was performed. Thereafter, the research defined risk and selected a risk model appropriate to Internet ordering systems. Risk was defined in relation to control objectives for computer systems, as defined by Watne and Turne (1990:308-309), and the risk model selected was the Access Model and the Path Context Model developed by Boshoff (1985, 1990).

Four layers in the "access path" of an Internet ordering system were selected for review, namely the HTML electronic form, HTTP, TCP/IP and CGI scripts. Risk factors identified varied according to the layers in the access path of an Internet ordering system. In the HTML layer, risk was influenced by the design elements of the electronic form. At the HTTP level, message header fields and server HTTP response codes impacted on risk. TCP/IP was also found to have header fields which affected risk. Finally, CGI scripts were found to be different to the other

layers. Risk factors at this level were found to be very dependent on the actual CGI scripts implemented.

The short dissertation opened additional areas for research. Of the three types of Internet EDI systems identified by The Gartner Group (1996b:4), only one was selected for research in this short dissertation. Clearly, opportunity exists for similar research to be conducted on the other Internet EDI implementations. Another area opened for research was the evaluation of risk in system-specific CGI scripts.

In conclusion, growth of electronic commerce on the Internet is being hampered by debate over taxation of goods and services sold over the Internet and by US government concerns over the exporting of encryption techniques (The Gartner Group, 1997b:1). However it appears that successful resolution of these issues is on the horizon (The Gartner Group, 1997b:1).

Chapter 1

1. Introduction

1.1 Background

"During the last 30 years, the Internet has evolved from a U.S. government network to today's international, commercially operated, government-subsidized network that is composed of many networks - and the Internet's evolution is not over yet" (The Gartner Group, 1996c:1). One particular aspect of this evolution is the extension of the Internet and Internet access to internal applications, making these applications available to external users (The Gartner Group, 1997a:1). "E-form templates are being used on the Web to turn passive browsers of enterprise Web sites into active users, providing feedback, requesting information and, with increased frequency, initiating transactions" (The Gartner Group, 1996b:16).

In South Africa, some organisations are already "Internet enabling" their internal applications. Examples of such enterprises are South African Airways (SAA) and Computicket. SAA has "Internet enabled" its airline ticket ordering/sales system, while Computicket has "Internet enabled" its ticket ordering and reservation system in the entertainment industry. Clearly, systems of this nature impact on the financial information presented by the enterprise. In the examples used, the sales and debtors (or cash) accounts are impacted by transactions concluded though the Internet.

1.2 Problem Description and Objective of Research

The South African Institute of Chartered Accountants (SAICA) has stated that an "auditor should obtain an understanding of the accounting systems sufficient enough to identify and understand...the accounting and financial reporting process, from the initiation of significant transactions and other events through to their inclusion in the financial statements" (SAAS 400, .15). SAICA goes on to recommend that the auditor should then make a preliminary assessment of control risk for each material balance or class of transactions (SAAS 400, .18).

Therefore, in instances where "Internet enabled" systems are the source of significant transactions, the auditor is required to obtain an understanding of the process through which these transactions are initiated and, eventually, are recorded in the financial statements. Furthermore, the auditor will need to make a preliminary assessment of control risk relating to these systems.

Due to the increasing sophistication of these computerised systems, auditors are forced to target computers as the subject of audits and to audit through computers (Watne & Turne, 1990:14). Consequently, auditors involved in the review of organisations with Internet ordering systems will have no choice but to target these systems in their audits. There may be certain instances where Internet ordering systems are the source of a material class of transactions, such as sales.

The literature survey conducted did not address risks in e-form ordering systems from an auditor's perspective. Therefore computer auditors at present do not have a model which assists them in evaluating risks in an Internet ordering system. With the trend of enterprises attempting to "Internet enable" their applications (The Gartner Group, 1997a:1), computer auditors will require a risk model which may be applied to such ordering systems.

The objective of this research is, therefore, to develop a risk model which may be applied to Internet-based ordering systems, as such a model has yet to be developed. The model should assist auditors in making an assessment of control risks in such systems, where orders of this nature have a material impact on sales transactions.

1.3 Scope, Limitations and Exclusions

This short dissertation will discuss the impact of Internet technology on ordering systems. The model presented will take account of the process from the point the customer places the order up to the point where the order is confirmed, or rejected, by the supplier through the ordering system itself.

Issues which are not dealt with in this research include the following:

- Payment mechanisms, such Secure Electronic Transactions (SET), through which the customer suitably compensates the supplier for goods or services received;
- The process through which orders on the ordering system are transferred to the financial accounting system;
- Various Internet control mechanisms, such as firewalls and proxies;
- The hardware used to implement this technology; and
- Platform-specific software, such as operating systems, and specific implementations of ordering application systems.

1.4 Definitions

Some essential terms need to be defined in order to appreciate the technology involved in Internet ordering systems.

1.4.1 Browser

The front-end to Internet and intranet services. Mosaic is the progenitor of today's familiar Web browsers. To the user, the Web browser application pulls down and displays hypertext information a page at a time. These pages, constructed in HTML, can contain embedded graphics and selected formatting primitives. They also contain links, expressed usually as underlined words or icons, the selection of which invokes an HTML page from anywhere on the Internet, allowing the user to jump from one information source to another with the simple click of a mouse. Some links point not to HTML pages but to files accessed via standardised Internet FTP links. Clicking on one of these will automatically copy a file to the user's device and, depending on the browser installed, may automatically invoke an application other than the browser, i.e., a local helper application, to act on the automatically downloaded file (The Gartner Group, 1996b:43).

1.4.2 Common Gateway Interface (CGI) Scripts

The interface behind Web servers through which programmers can construct scripts to invoke other server-based processes, pass information to them and receive results back (The Gartner Group, 1996b:43).

1.4.3 EDI (Electronic Data Interchange)

The electronic transfer of preformatted business documents, such as purchase orders and bills of lading, between trading partners (The Gartner Group, 1996b:44).

1.4.4 E-form (Electronic Form)

An automated and interactive template for the capture, processing, display and output of defined sets of business data (The Gartner Group, 1996b:44).

1.4.5 Electronic commerce

Marketing products and services, handling inquiries, taking orders, and conducting customer-support dialogs and surveys using an evolving array of technical tools to link multiple enterprises. Key electronic commerce tools include the use of e-forms, EDI, bulletin boards, messaging, shared online databases and electronic payments. Electronic commerce also includes custom applications that enterprises are building - or are having built - to link up with their trading partners (The Gartner Group, 1996b:44).

1.4.6 HyperText Markup Language (HTML)

The WWW's document description language standard (The Gartner Group, 1996b:45).

1.4.7 HTTP (Hypertext Transport Protocol)

A key application protocols standard for the Internet. Between the WWW browser client and the Web server, communications occur over a session-less communications protocol known as the HTTP (The Gartner Group, 1996b:45).

1.4.8 Internet

The Internet is an informal, global network of tens of thousands of machines which support certain Internet Engineering Task Force and WWW Consortium promulgated standards and derivatives thereof. Key standards are platform independent and include node addressing and routing, network transports (e.g., TCP/IP), application protocols (e.g., FTP, HTTP, POP3 and IMAP), document definition languages (e.g., HTML) and hierarchical storage standards (e.g., Gopher and NNTP) (The Gartner Group, 1996b:45).

1.4.9 World Wide Web (WWW)

The multimedia portion of the Internet (The Gartner Group, 1996b:48).

1.5 Research Methodology

The approach used will be a three-step process. Firstly the technological aspects and operation of such Internet ordering systems are introduced and discussed in Chapter 2. Specifically, the various components of such a system have been identified and the operational aspects of these components have been discussed in the chapter. Parameters which may potentially impact on risk are discussed in detail. This was done through a survey of available literature.

Chapter 3 addresses the second step in the research process. An appropriate risk model which may be applied to Internet ordering systems is introduced. The

application of the model to such a system is discussed in the chapter. The risk model has been identified from authoritative literature on the subject of auditing.

Chapter 4 evaluates the application of the risk model to the Internet ordering system technology. One of the results of this is the identification of risks in Internet ordering systems as they impact on auditors. The results are presented in a matrix, evaluating the individual risk components against each of the components in Internet ordering systems.

1.6 Summary of Results

Several components of Internet ordering systems have been identified in Chapter 2. These include the following:

- Client browers and HTML e-forms;
- HTTP;
- TCP/IP; and
- CGI scripts and supplier ordering systems.

HTML e-forms are used to design the order forms through which Internet ordering systems are implemented (Ragget, 1997). HTTP and TCP/IP are communication protocols between the client and the supplier's server (The Gartner Group, 1996b:45), while CGI scripts are used to interface between the supplier's Web server and the internal ordering system (NCSA, University of Illinois at Urbana Champaign, n.d. a). Each of these components have been found to have parameters which impact upon the functionality thereof. These parameters and the interaction between the components of an Internet ordering system are discussed in detail in Chapter 2.

Risk is defined in Chapter 3 and is based on the definition given by Watne and Turne (1990:308-309). The definition of risk considers the following control objectives:

- Transaction authorisation;
- Error detection, correction and resubmission;
- Security of data and records;
- Distribution of output;
- Accuracy of input, processing and output;
- Uniqueness of transactions; and
- Reasonableness of processing results.

(Watne and Turne, 1990:308-309)

An appropriate risk model, namely the Path Context and Access model as developed by Boshoff (1985, 1990), is selected for the purposes of this research.

In Chapter 4, the selected risk model is applied to the technology infrastructure introduced in Chapter 2, to identify risks in an Internet ordering system. The technology components are mapped against the control objectives in a matrix, and where a control objective has not been met, a risk exists for that control objective. The relevant section where the technology component has been discussed is also presented in the matrix. The matrix developed in Chapter 4 presents a risk model which may be used by computer auditors to evaluate risks in an Internet ordering system.

1.7 Conclusion

The introduction to Internet ordering systems and the operational aspects thereof that has been provided in Chapter 2 serve to give the auditor an understanding of such systems. The risk model developed through this research highlights risk factors which the auditor needs to bear in mind when faced with such a system in practice. The results presented in this short dissertation are by no means the only factors that an auditor needs to consider in such systems. This short dissertation has not addressed risk in other areas, such as payment receipt mechanisms, the summarisation of Internet orders and the posting of sales into the general ledger, and the impact of firewalls on such technology. These are areas which may require further research and it is hoped that this short dissertation will serve as a starting point for such research.

Chapter 2

2. Internet Ordering System Technology

2.1 Objectives

In order to identify EDI Internet risks in an ordering system, key terms need to be defined and important concepts need to be explained. The broad objectives of Chapter 2 are to introduce the technological concepts used in such a system, and to provide a basic understanding of the workings of an Internet ordering system. Chapter 3 intends to select a risk model which may be applied to an Internet ordering system. Chapter 4 will integrate the technological concepts introduced in Chapter 2 with the risk model selected in Chapter 3, to present a risk model in an Internet ordering system.

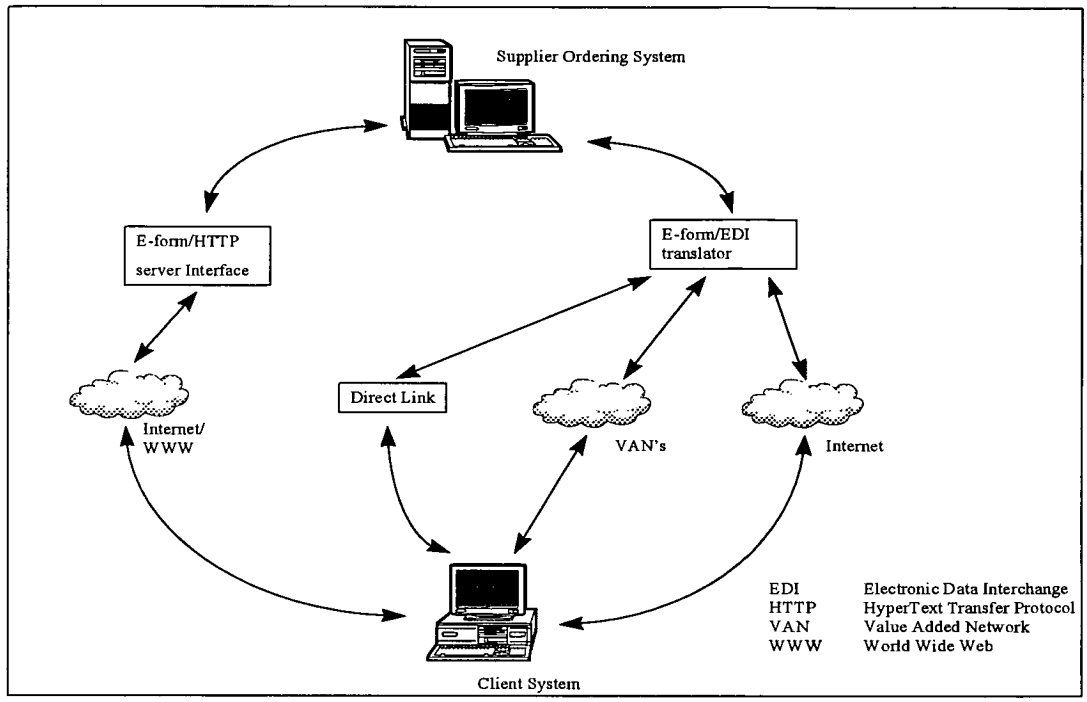
2.2 Nature of Literature Survey

The literature survey conducted comprised Internet documents and published literature. Information on the technological aspects of the subject was obtained through a survey of the available literature on the Internet. Various educational institutions with Internet sites were found to have helpful literature on the subject. A literature survey of published material was conducted to identify the various risk-related issues. The material comprised publications from the South African Institute of Chartered Accountants and other authoritative publications from various authors on the subject of auditing.

2.3 Scope, Limitations and Exclusions

The literature survey has identified implementations of ordering systems over the Internet. The Gartner Group (1996b:4) has identified two distinct types of e-forms. Figure 2.1 illustrates these.

Figure 2.1: Implementing EDI over the Internet (The Gartner Group, 1996b:4)



In figure 2.1, the first type of e-form implementation is where e-forms are implemented through a HyperText Transfer Protocol (HTTP) server interface. This implementation makes use of HyperText Markup Language (HTML) to design the form, and the Internet as a communication channel between the customer and the supplier. The alternative implementation is where an e-form is implemented through an e-form application product. Such a product provides for a data translator facility between the e-form and the local EDI system. The supplier and the customer have a choice between three types of communication channels to effect data transfer. Firstly, a direct link may be established between the supplier and the customer. Secondly, the parties may use a Value Added Network to store and forward EDI messages. Thirdly, the Internet may be used as a communications medium to transport EDI messages.

Clearly, discussion of all of the three implementations of Internet ordering systems would be extensive, as each is unique in the nature of its implementation and the risks involved therein. "Although EDI translation is an appealing solution for routing e-form data between trading partners, it does not appeal to the emerging population of Internet users and small enterprises that want to use e-forms to

interact and do business with enterprises at lower costs. The sudden and spectacular rise in the popularity of the Web has prompted thousands of businesses to flock to it" (The Gartner Group, 1996b:3). Thus, the first method of implementation of an ordering system has been selected for review in this short dissertation.

The implementation of an ordering system over the Internet involves several technologies. This short dissertation is limited to discussing the following:

- The use of HTML to implement forms;
- The transportation of the forms over the Internet through HTTP and TCP/IP;
- The use of Common Gateway Interface (CGI) scripts to translate order information from an HTML format to a data format compatible with the supplier's ordering system; and
- The return of information from the CGI script on the supplier's server to the client, in an HTML format.

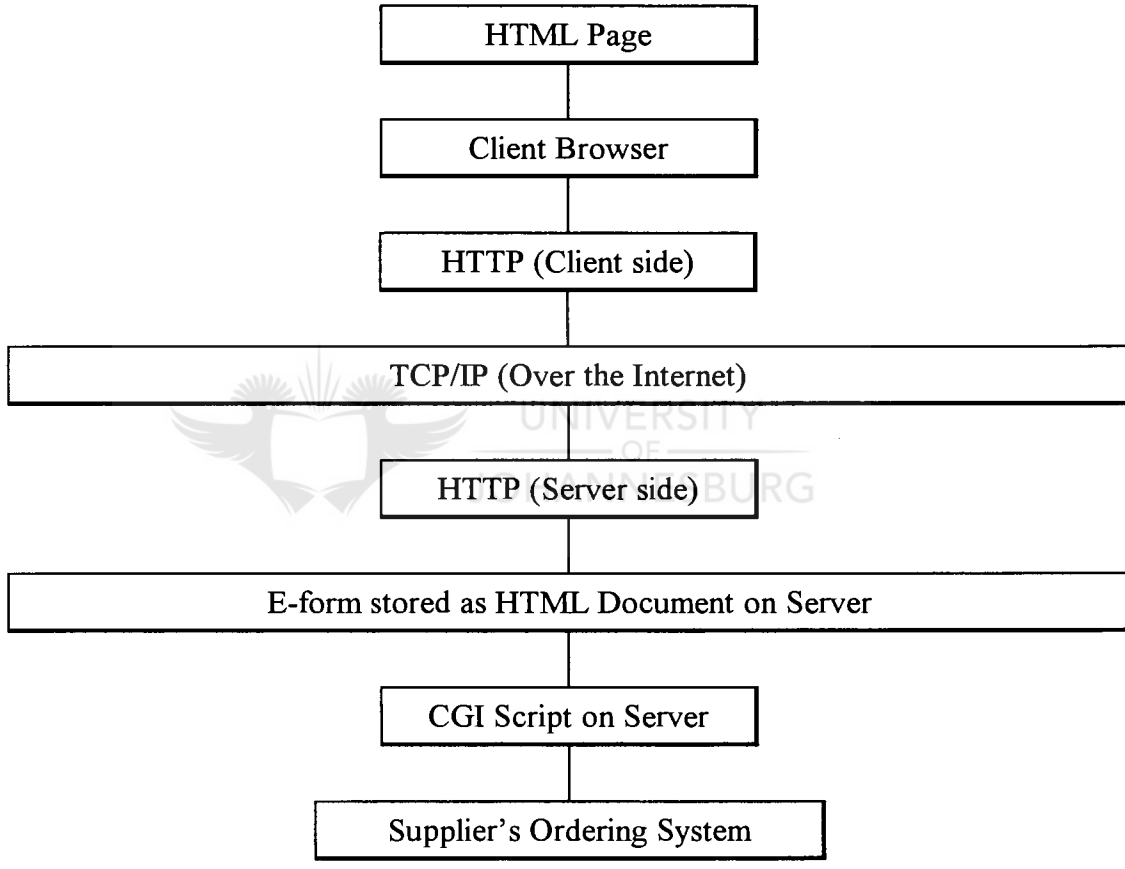
2.4 Introduction to Internet Ordering Systems

In section 1 of Chapter 1 of this short dissertation, it was observed that at least two companies in South Africa have implemented Internet ordering systems. In order to understand how these systems have been implemented, the process through which an order is placed and accepted through these systems is introduced in this section. The various components in Internet ordering systems are briefly introduced and the interaction between them is discussed.

Section 4.6 of Chapter 1 describes HTML as the document description language of the Internet. It is used to define e-forms (Ragget, 1997). These forms may be used to collect order information (Grobe, 1996). The client browser uses HTTP to display the HTML document, or e-form (Fielding, Irvine, Gettys, Mogul, DEC, Frystyk, Berners-Lee & MIT/LCS, 1997:36). HTTP is "a key application protocols standard for the Internet. Between the WWW browser client and the Web server,

communications occur over a session-less communications protocol known as the HTTP" (The Gartner Group, 1996b:45). HTTP resides on top of TCP/IP (The Gartner Group, 1996a:35). TCP/IP is the network protocol used on the Internet (The Gartner Group, 1996b:45). Finally, the CGI script on the server side provides the interface between the Web server and the supplier's ordering system (NCSA, University of Illinois at Urbana Champaign, n.d. a). These concepts are illustrated in figure 2.2.

Figure 2.2: Framework of Internet ordering systems



The various components of Internet ordering systems and their interaction have been introduced. Each of the components will now be reviewed in detail.

2.5 Client Browser and HTML Formatted E-Form

The ordering process is initiated by the customer requesting an HTML document from the supplier to view on his or her browser. The request is initiated through

the client's browser software (Grobe, 1996). The HTML document requested is an order form formatted as an e-form and resides on the supplier's server. The browser software sends a request for the web document to the supplier's server. It also communicates the HTTP version it will accept to the supplier's server (Grobe, 1996). This process is discussed in more detail in section 6.1.

The server returns the page (in this case, an electronic order entry form) to the customer in HTML format. The client's web browser translates the standard HTML information into a format which is presented on screen to the client (Grobe, 1996).

An HTML document is a standard text (ASCII) document with special 'tags' defining various elements within the web document. These tags are enclosed between angle brackets ('<>') to distinguish them from normal text. The tag '<html>' on the first line of an ASCII file identifies the file as an HTML document. Tags are designed to be used either on their own or in pairs. An example of a tag used in a pair is the one used to define the title of an HTML document. Thus one would define a title of an HTML document by enclosing it between the tags '<TITLE>' and '</TITLE>' (NCSA, University of Illinois at Urbana Champaign, 1996).

Tags are the vehicle through which electronic forms (also referred to as e-forms) are implemented on the client side. These e-forms are used to provide an input area for the client to capture order information. All forms are identified by and enclosed between the tags '<FORM>' and '</FORM>' (Ragget, 1997). Two parameters follow the '<FORM>' tag. Firstly, the method of transferring the e-form data is defined through the 'METHOD' parameter. This is discussed in section 6.1 of this chapter. The other parameter is 'ACTION'. This tells the server what to do with the information received from the client (Ragget, 1997). The 'ACTION' parameter points to the CGI script that will process the form information (Ragget, 1997). The CGI script is the gateway between the Internet-based e-form and the supplier's order processing system (The Gartner Group, 1996b:43). Its functionality is discussed in section 8 of this chapter.

Various types of input are available within an electronic form, such as (Ragget, 1997):

- Plain text;
- Multi-line entry;
- Option menus;
- Scrolled lists;
- Radio buttons;
- Check boxes;
- Password entry;
- Reset; and
- Submit.

Each of the types of input form is identified by the tag '<INPUT TYPE =' followed by a parameter (Ragget, 1997). The input types will now be discussed.

2.5.1 Plain text and Multi-line Entries

Plain text is normally a single line of text which the client may complete (Ragget, 1997). It may be used, for example, as an input for a customer code. Multi-line entries are similar to plain text input except that the client may complete several lines of information (Ragget, 1997). This may be used by the client to complete his or her delivery address, or optional information.

2.5.2 Option Menus and Scrolled Lists

Option menus are a form of input whereby the client selects items from a menu list of several items (Ragget, 1997). An example of where this may be used on a form is where the supplier wishes to provide the customer a list of valid products sold. Another feature of option menus is that the customer may be forced to select a single item or multiple items through parameters in the HTML form; such

parameters are implemented through tags (Ragget, 1997). Scrolled lists are identical to option menus, except that the size of the window displaying the list is limited and the user can scroll through the list of items (Ragget, 1997). As with the option menus, the customer may be forced to select a single item from the list or the customer may select multiple items from such a list (Ragget, 1997).

2.5.3 Radio Buttons and Check Boxes

Radio buttons make a user select one item from a selection and use of radio buttons is appropriate where the number of options a user is presented with is limited (Ragget, 1997). Thus, if a user is required to select one option from a host of options, an option menu or a scrolled list may be appropriate. On the other hand, where a user is required to select one item from a range of, say, three choices, use of radio buttons is suggested. Check boxes are similar to radio buttons, except that the user may select more than one option, whereas with radio buttons the user is forced to make a single choice (Ragget, 1997).

2.5.4 Password Entries

Password entries are identical to plain text entries with one exception, namely that the customer does not see what is being typed onto the screen. As the user inputs data, the client browser "echoes characters like '*' to hide the text from prying eyes" (Ragget, 1997). Input of this type may be used to authenticate the identity of the client on the supplier's server.

2.5.5 Reset and Submit Buttons

The last two input types are different from the ones discussed so far. The "reset" parameter "defines a button that users can click to reset form fields to their initial state when the document was first loaded. ...Reset buttons are never sent as part of the form's contents" (Ragget, 1997). The submit parameter defines a button which the user may click on to transmit the e-form data to the supplier's server.

Having discussed the various types of input forms available and the mechanism through which a client may initiate the transmission of the e-form data to the server from the browser, it is appropriate to consider the process through which the electronic order information is communicated to the server.

2.6 The HyperText Transfer Protocol (HTTP)

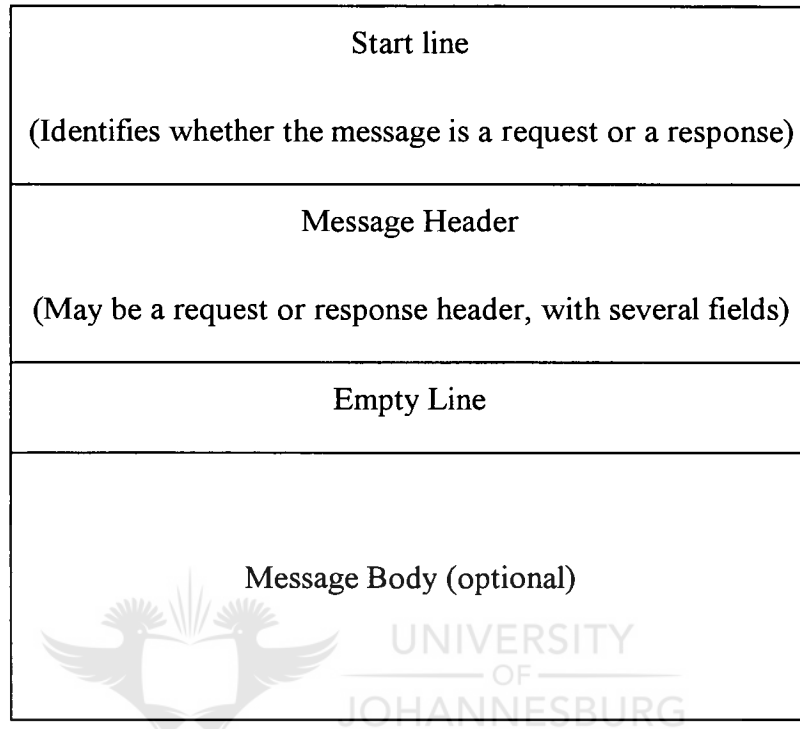
HTTP is "an application-level protocol...[which is]...used as a generic protocol for communication between user agents and proxies/gateways to other Internet systems" (Fielding, et al., 1997:1). It is a request response protocol in which a client sends a request to the server and the server responds with a message (Fielding, et al., 1997:11).

The functions of the application, presentation and session layers will be briefly discussed to place the function of HTTP in the context of the OSI Reference model. The application layer is the application itself which uses information received from another computer. The presentation layer controls the encoding of data types to facilitate exchange thereof between different systems. The session layer controls the exchange of messages in the transport layer (Santifaller, 1994:8). In comparing HTTP with the OSI model, an observation may be made that HTTP encompasses the application, presentation and session layers of the OSI Reference model. This is because TCP/IP architects view "the functions of the presentation and session layers as part of the application layer" (Santifaller, 1994:14). Thus HTTP "runs on top of TCP/IP" (The Gartner Group, 1996a:35), serving the functionality of the three layers.

HTTP messages may be either requests from the client to the server or responses from the server to the client (Fielding, et al. 1997:30). A more detailed discussion of HTTP request and response messages follows. An understanding of such messages will aid in grasping the communication process between the client and the server.

Fielding, et al. (1997:30) have defined the generic structure of an HTTP message. This is illustrated in figure 2.3:

Figure 2.3: Generic structure of HTTP messages (Fielding, et al., 1997:30)



2.6.1 Request Messages

Request messages are messages that the client sends to the server (Fielding, et al. 1997:34). The generic structure of the request start line is:

METHOD [space] Web Page Requested [Space] HTTP Version (Fielding, et al. 1997:34)

Two methods are of interest for the purposes of e-forms, namely the 'GET' method and the 'POST' method. The former is used to retrieve the contents of a web page from the server. The web page retrieved is identified by 'Web Page Requested' (Fielding, et al., 1997:50).

The latter method is used to provide "a block of data, such as the result of submitting a form, to a data handling process" (Fielding, et al., 1997:51). In the case of the 'POST' method, the 'Web Page Requested' is the name of the web page on the server which will control the processing of the block of data sent by the client (recall the 'ACTION' parameter in section 5). The 'HTTP version' is the protocol version accepted by the client.

Note that the 'GET' method may also be used to provide data submitted on a form to a data handling process. This is achieved by appending a '?query _string' at the end of the 'Web Page Requested' (University of Toronto, n.d.).

2.6.2 Request Message Header Fields

The request header is used to transfer additional information about the request, in a manner similar to the passing of parameters to a program. Request headers are composed of fields. Several request header fields exist for the 'GET' and 'POST' methods. However, only those which could potentially impact on risk in an Internet ordering system are discussed. These include:

- Accept-Charset;
- Accept-Encoding;
- Accept-Language;
- Accept-Authorisation;
- From;
- Proxy-Authorisation;
- Range:
  - ⇒ Byte Ranges;
  - ⇒ Range Retrieval Request;
- User Agent.

(Fielding, et al., 1997:97-134)

The 'Accept-Charset' field is used by the client to indicate the acceptable character sets to the server. The 'Accept-Encoding' field is used to indicate the message encoding methods recognisable by the client (Fielding, et al., 1997:97). The 'Accept-Language' field is used to restrict "the set of natural languages that are preferred as a response to the request" (Fielding, et al., 1997:98). In the event that the server returns a '401' response code (i.e. that the user is unauthorised), the client may include an 'Authorisation' field, enabling him or her to access the resource (web page) on the server (Fielding, et al., 1997:100). The 'From' field provides the server with the Internet e-mail address of the client. This may be used by the server for logging purposes (Fielding, et al., 1997:118). The 'Proxy-Authorisation' field identifies the client to a proxy which requires authentication. It is used when the server returns a 'Proxy-Authenticate' response header field (Fielding, et al., 1997:127). 'Byte ranges' are used to specify the information to be extracted from the server. Thus if the web page is viewed as a collection of bytes, 'Byte Ranges' will specify the sequence of bytes which is to be extracted from the server (Fielding, et al., 1997:128-130). The 'Range Retrieval Request' is similar in functionality to the 'Byte Ranges' field, except that it differs in its format (Fielding, et al., 1997:130). The 'User-Agent' request header field is used to supply the server with information about the client originating the request. "This [information] is [used] for statistical purposes, the tracing of protocol violations, and automated recognition of user agents for the sake of tailoring responses to avoid particular user agent limitations" (Fielding, et al., 1997:134).

2.6.3 Response Messages

The HTTP request message has been introduced and described briefly. To complete the process, the HTTP response message must be addressed. The generic format of an HTTP message, as depicted in figure 2.3, can also be applied to the response message. The start line of the message on the server side is referred to as the status line, and it is structured as follows (Fielding, et al. 1997:38):

PROTOCOL VERSION [Space] STATUS CODE [Space] Description

The 'PROTOCOL VERSION' is the HTTP version the response is formatted in (Fielding, et al. 1997:39). The 'STATUS CODE' is discussed in detail in section 6.4.

The third part of the status line, the 'Description', is a textual description of the status code returned by the server, meant to be understood by humans (Fielding, et al., 1997:39).

2.6.4 Response Message Status Codes

The 'STATUS CODE' is a numeric value which returns the status of the request (Fielding, et al. 1997:39). The details of all the possible response codes will not be discussed as this is available in the literature (Fielding, et al., 1997:53-65). However, the codes which are considered relevant in this research are presented in table 2.1

Table 2.1: HTTP response message status codes (Fielding, et al., 1997:53-65)

Code	*Description*
100	*Continue* The client may continue with the request. The initial part of the request has been received and has not yet been rejected by the server.
200	*OK* The request has succeeded.

Table 2.1: HTTP response message status codes (Continued)

Code	*Description*
201	*Created* The request has been fulfilled and has resulted in a new resource being created. The newly created resource is given by a Location header field.
202	*Accepted* The request has been accepted for processing, but the processing has not been completed. The message content returned with this response indicates the current status of the client's request and either a pointer to a status monitor or some estimate of when the user can expect the request to be fulfilled.
205	*Reset Content* The server has fulfilled the request and the client should reset the e-form. This response is primarily intended to allow input for actions to take place via user input, followed by a clearing of the form in which the input is given so that the user can easily initiate another input action.
305	*Use Proxy* The e-form must be accessed through the proxy given by the Location field. The Location field gives the address of the proxy. The client is expected to repeat the request via the proxy.
400	*Bad Request* The request could not be understood by the server due to malformed syntax. The client should repeat the request after fixing the syntax.

Table 2.1: HTTP response message status codes (Continued)

Code	*Description*
401	*Unauthorised* The request requires user authentication. The response must include a WWW-Authenticate header field with the authentication parameters applicable to the requested resource.
403	*Forbidden* The server understood the request, but is refusing to fulfil it. Authorisation will not help and the request should not be repeated. This status code is commonly used when the server does not wish to reveal exactly why the request has been refused, or when no other response is applicable.
407	*Proxy Authentication Required* This code is similar to 401 but indicates that the client must first authenticate himself or herself with the proxy. The proxy must return a Proxy-Authenticate header field containing a challenge applicable to the proxy for the requested resource. The client may repeat the request with a suitable Proxy-Authorisation header field.
408	*Request Timeout* The client did not produce a request within the time that the server was prepared to wait. The client may repeat the request.

Table 2.1: HTTP response message status codes (Continued)

Code	*Description*
409	*Conflict* The request could not be completed due to a conflict with the current state of the resource (e-form document). This code is only allowed in situations where it is expected that the user might be able to resolve the conflict and resubmit the request. The response body should include enough information for the user to recognise the source of the conflict.
411	*Length Required* The server refuses to accept the request without a defined Content-Length. The client may repeat the request if it adds a valid Content-Length header field containing the length of the message-body in the request message.
414	*Request-URI Too Long* The server is refusing to service the request because the requested web page (through the 'GET' method) is longer than the server is willing to interpret. This rare condition is only likely to occur when a client has improperly converted a 'POST' request to a 'GET' request with long query information.
500	*Internal Server Error* The server encountered an unexpected condition which prevented it from fulfilling the request.

Table 2.1: HTTP response message status codes (Continued)

Code	*Description*
501	*Not Implemented* The server does not support the functionality required to fulfil the request. This is the appropriate response when the server does not recognise the request method and is not capable of supporting it for any resource (in this case the web page requested).
502	*Bad Gateway* The server, while acting as a gateway or proxy, received an invalid response from the upstream server it accessed in attempting to fulfil the request.
503	*Service Unavailable* The server is currently unable to handle the request due to a temporary overloading or maintenance of the server. The implication is that this is a temporary condition which will be alleviated after some delay. If known, the length of the delay may be indicated in a Retry-After header. If no Retry-After is given, the client should handle the response as it would for a 500 response.
504	*Gateway Timeout* The server, while acting as a gateway or proxy, did not receive a timely response from the upstream server it accessed in attempting to complete the request.

Table 2.1: HTTP response message status codes (Continued)

Code	*Description*
505	*HTTP Version Not Supported* The server does not support, or refuses to support, the HTTP protocol version that was used in the request message.

2.6.5 Response Message Header Fields

As was the case with request messages, response messages also contain header fields. These include the following:

- Location;
- Proxy-Authenticate;
- Public;
- Retry-After;
- Warning; and
- WWW-Authenticate.

(Fielding, et al., 1997:125-139)

Again, the list of header fields is not complete; only those response header fields which may impact on risk on an Internet Ordering system are discussed.

'Location' is a new web page which was created on the server side as a result of a request from a client (Fielding, et al., 1997:125). This response header field is used in conjunction with CGI scripts. This will be discussed in section 8. The 'Proxy-Authenticate' field is used to indicate to the client that it must authenticate itself with the server. This field is included when the server returns a status code of 407, defined as 'Proxy Authentication Required' (Fielding, et al., 1997:127). 'Public' is a response header field identifying the methods accepted by the server (Fielding, et al., 1997:127-128). In certain instances the server may not be available. This is

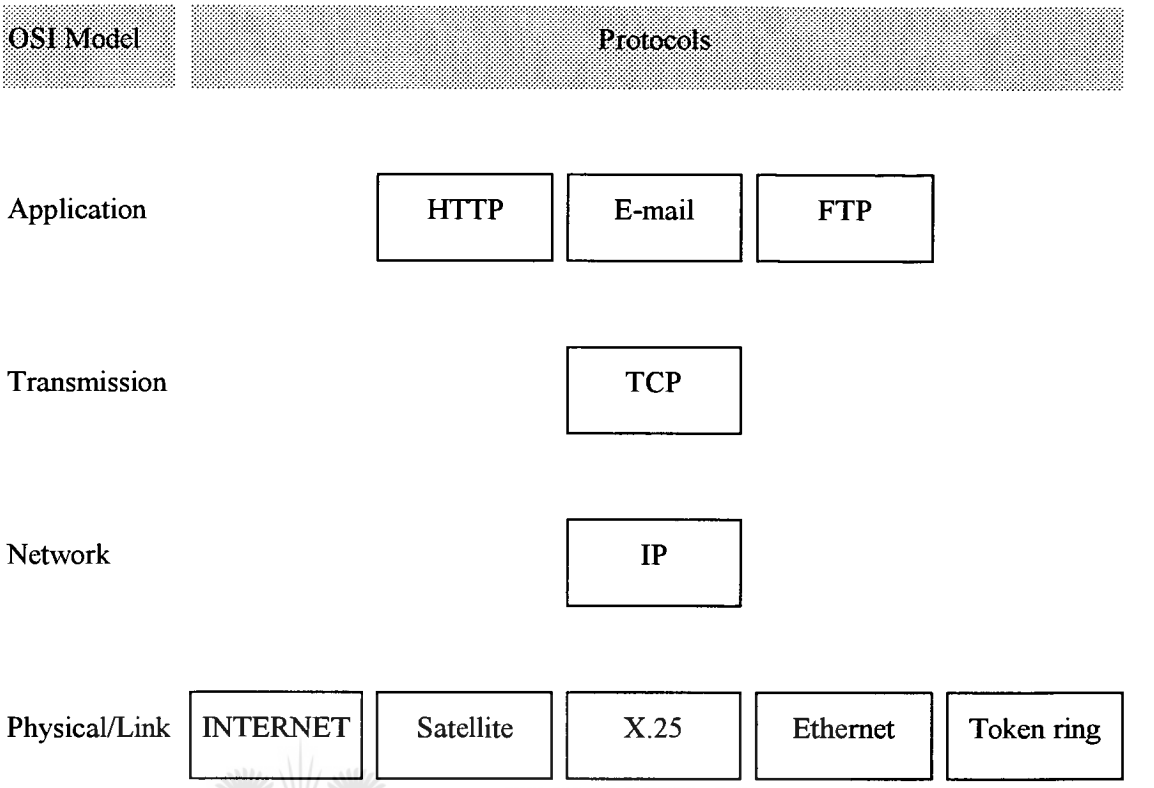
indicated through the status code 503 in the response message status line (Fielding, et al., 1997:64). Should this be the case, the server may return a 'Retry-After' response header field informing the client the time period to wait before retrying to access the server (Fielding, et al., 1997:131). "The Warning response-header field is used to carry additional information about the status of a response which may not be reflected by the response status code" (Fielding, et al., 1997:137). The 'WWW-Authenticate' response header field is used when a 401 status code is returned (Fielding, et al., 1997:139). This status code reflects that "the request requires user authentication" (Fielding, et al., 1997:60). The response header field contains the authentication parameters required from the client, by the server (Fielding, et al., 1997:139).

HTTP, its processes and parameters have been briefly introduced. Bearing in mind that it functions as the application, presentation and session layers of the OSI Reference model, the next step is to consider the lower layers (and details thereof) of the OSI Reference model.

2.7 The Transmission Control Protocol / Internet Protocol (TCP/IP)

The purpose of the network layer is to transport messages between communication partners, ensuring that data flow between communication partners is controlled and that data is not corrupted. TCP is a transport protocol. The network layer establishes a virtual path between communication partners. IP is an example of a network protocol (Santifaller, 1994:8). It should be noted that TCP/IP requires independence from the layers below it. This is achieved through the link and physical layers being treated as subnetworks which are implemented in a closed manner on intelligent network controller cards, for example Ethernet. Therefore, "it is possible to operate TCP/IP over X.25 or any other network" (Santifaller, 1994:15). This is the manner in which one of the goals set by TCP/IP architects is achieved, namely the goal of independence from the underlying network technology (Santifaller, 1994:13). This is presented in figure 2.4.

Figure 2.4: Independence of TCP/IP from underlying OSI physical and link layers (Santifaller, 1994:13)



2.7.1 Transmission Control Protocol (TCP)

TCP's main task is to ensure the reliable transportation of data through a network. Messages are structured with headers and data segments (Santifaller, 1994:31-32). As was done with HTTP, some of the TCP header fields are discussed to give an appreciation of the structure and function of TCP.

The TCP header comprises the following fields (Santifaller, 1994:32):

- Source port;
- Destination port;
- Sequence number;
- Acknowledgement number;
- Data offset;

- Flags:
    - ⇒ ACK;
    - ⇒ PSH;
    - ⇒ RST;
    - ⇒ SYN;
    - ⇒ FIN;
- Window size;
- Checksum; and
- Options.

As explained earlier, all header fields are not being introduced and discussed; only those which may have an impact on risk are outlined.

The source and destination port numbers are used to establish a TCP connection between communication partners (Santifaller, 1994:35-36). The sequence number indicates the starting position of data within the data stream and is determined by the sender. The acknowledgement number is similar to the sequence number except that it is determined by the recipient and is based on the sequence number. The recipient adds the number of data bytes received to the sequence number to obtain the acknowledgement number (Santifaller, 1994:33). In summary, the sender's sequence number represents the recipient's acknowledgement number, and vice versa.

The data offset determines the length of the TCP header (Santifaller, 1994:33). Flags are used to trigger actions in TCP. The flags and their meaning are (Santifaller, 1994:33):

- ACK: Acknowledgement number is valid;
- PSH: Data received in this segment should be passed on to the application This flag is used to indicate to the recipient that an entire message (comprising several data segments) has been successfully sent by the sender;

- RST : This indicates a reset of the connection due to an invalid segment;
- SYN : Connection set up request; and
- FIN : The connection is shut down from one side and data flow from the sender of this flag has ended.

The window size indicates the number of bytes that the recipient is able to store in its buffers. The checksum field is used to calculate and store a checksum based on the TCP and the IP headers. It prevents incorrect forwarding of messages by IP and detects data loss (Santifaller, 1994:34). One of the options which TCP may have is to limit the maximum size of a data segment. This is indicated in the options field (Santifaller, 1994:35).

2.7.2 TCP Connection Process

Figure 2.5 illustrates the connection, data exchange and connection shutdown process.

Figure 2.5: Connection, data exchange and connection shut down over TCP/IP (Santifaller, 1994:39-40)

Connection

Step	Client	TCP Header fields and flags	Server
Connection	→	Sequence = 100, Flags = SYN	
		Sequence = 300, Acknowledge = 101, Flags = SYN,ACK	←

Figure 2.5: Connection, data exchange and connection shut down over TCP/IP (Continued)

Step	Client	TCP Header fields and flags	Server
	→	Sequence = 101, Acknowledge = 301 Flags = ACK	

Data exchange

Step	Client	TCP Header fields and flags	Server
Data Exchange	→	Sequence = 101, Acknowledge = 301, Flags = ACK, No. of data bytes sent = 5	
		Sequence = 301, Acknowledge = 106, Flags = ACK, No. of data bytes sent = 10	←
	→	Sequence = 106, Acknowledge = 311, Flags = ACK	

Figure 2.5: Connection, data exchange and connection shut down over TCP/IP (Continued)

Connection shut down

Step	Client	TCP Header fields an flags	Server
Connection Shutdown	→	Sequence = 106, Flags = FIN, ACK	
		Sequence = 311, Acknowledge = 107, Flags = ACK, No. of data bytes sent = 5	←
	→	Sequence = 107, Acknowledge = 316, Flags = ACK	
		Sequence = 316, Acknowledge = 107, Flags = FIN, ACK	←
	→	Sequence = 107, Acknowledge = 317, Flags = ACK	

Figure 2.5 assists in understanding the use of the sequence and acknowledgement numbers, connection set-up flag, acknowledge flag, and the connection shutdown flag.

2.7.3 Internet Protocol (IP)

Having understood the workings of TCP, it is necessary to discuss the Internet Protocol (IP) and its role in an Internet ordering system. Santifaller (1994:19) recognises the main tasks of IP as being the addressing of computers and the fragmentation of packets; IP does not contain functions for end-to-end message reliability or for flow control. As with the structure of HTTP and TCP messages, IP messages also include header fields. These include (Santifaller, 1994:20-23):

- Version number;
- Length;
- Type of service;
- Packet length;
- Identification;
- Flags:
  - ⇒ More fragments;
  - ⇒ Don't fragment;
- Fragment offset;
- Time to live;
- Transport protocol;
- Header checksum; and
- Source and destination address.

The version number specifies the version of IP being used. The length header field contains the length of the IP header (Santifaller, 1994:20). The type of service field can be used to control the throughput and reliability with which the message is processed. Packet length points out the length of the IP packet, including the header. The identification number is a unique number which identifies the datagram to which the IP packet being sent belongs (Santifaller, 1994:21). The flags field contains two flags, namely: whether or not more fragments in the current data sequence can be expected (MF flag); and whether or not the current packet can be

fragmented (DF flag). The fragment offset field specifies the position of the data contained in the packet relative to the whole message data being received. It is used in re-assembly of the fragments. The time to live field contains the life span of the packet, i.e. the amount of time the packet may be kept alive before being discarded from the network (Santifaller, 1994:22). The transport protocol field contains a value which identifies the overlying transport protocol. A value of 6 specifies that TCP is the overlying transport protocol (Santifaller, 1994:22-23). The header checksum field contains a calculated checksum for the IP header fields. Note that the checksum does not include data field contents in its calculation. The source and destination addresses fields contain the addresses for the source and destination of the IP packets.

2.7.4 IP Packet Fragmentation and Reassembly

"In order to transmit packets over any type of network, IP must be capable of adapting the sizes of its datagrams to each network" (Santifaller, 1994:28). Fragmentation is used to achieve this goal. It is the process whereby a sender is able to fragment packets, each fragmented packet with its own IP header, and the receiver is able to reassemble fragmented messages. The identification number, flags and fragment offset header fields play a crucial role in the fragmentation and reassembly process. This is illustrated in figure 2.6.

Figure 2.6: The fragmentation process (Santifaller, 1994:28-29)

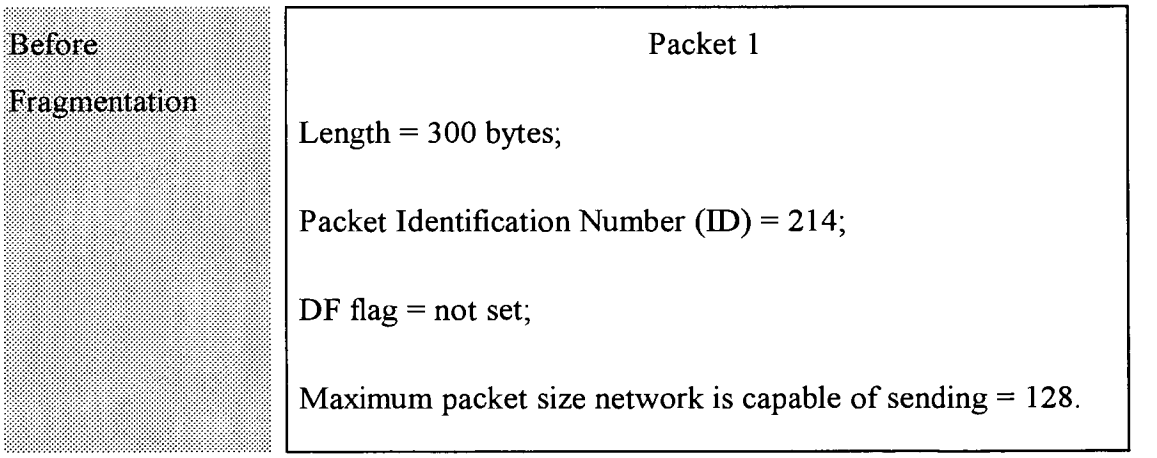


Figure 2.6: The fragmentation process (Continued)

After Fragmentation	Fragment 1	Fragment 2	Fragment 3
	Length = 124;	Length = 124;	Length = 112;
	Offset = 0;	Offset = 13;	Offset = 26;
	ID = 214;	ID = 214;	ID = 214;
	MF flag = set.	MF flag = set.	MF flag = not set.

It should be noted that the offset is calculated in units of 8 bytes. This is why the offset is not incremented by 124 from fragment 1 to fragment 2, and by the same amount from fragment 2 to fragment 3.

This section has introduced TCP/IP and the functionality thereof. The functionality of HTTP, the client browser software and HTML have also been discussed. A brief review of the roles of the various components in the ordering process is warranted as it will summarise what has been discussed so far.

The customer launches the web browser software. For the purposes of this short dissertation, it is assumed that the order e-form web page is the first page the client browser attempts to access (i.e. the 'start-up' page). The client browser requests the web page through an HTTP 'GET' request. The request is translated into TCP. TCP ensures that the request is reliably transported through the network to its destination. TCP initiates connection with the supplier's server. It passes on the connection request to IP. The IP's function is to manage the addressing function, to fragment the request on the client side to suitable packet sizes (which the network is able to carry) and reassemble the fragments on the server side. On the server side TCP will 'handshake' with the client TCP to establish a connection and initiate data transfer. Once this is achieved, data transfer of the client browser's HTTP request message occurs.

HTTP on the server side accepts the request and processes it. The result is an HTTP response message being sent via TCP/IP (as described earlier). The response message includes the HTML e-form stored on the server. The e-form is received by the client browser and displayed on screen.

The customer completes the e-form (the order) and clicks on the submit button. Parameters within the e-form specify the method of e-form transfer to be used by the client browser (through 'METHOD = GET' or 'METHOD = POST') and the script that will be used by the server to process the form (through 'ACTION = scipt_name', where 'script_name' is the name of the script residing on the server side which will process form data). This process forms part of the HTTP layer.

The e-form data is transferred via TCP/IP. Once the order is received, it is processed and processing results are returned to the client (this is discussed in section 8). At the end of the exercise, the customer may close the client browser software. This will result in the client TCP initiating a connection shutdown with the server TCP. Once the connection is shut down, the process is complete.

The next steps in the process are the order receipt, order processing in the ordering system and response to the client. These are discussed in section 8.

2.8 Order Receipt and Processing: Common Gateway Interface (CGI) Scripts

"The Common Gateway Interface (CGI) is a standard for interfacing external applications with information servers, such as HTTP or web servers" (NCSA, n.d. a). It therefore acts as a conduit converting HTML e-forms to a format compatible with the supplier's ordering system and then passing on this data to the ordering system. Once this conversion is complete, the ordering system may process the information received from the client.

The e-form information transfer is initiated by the client through clicking on the 'submit' button on a form. Once this happens, the server activates the CGI script indicated by the 'ACTION' parameter (refer to section 5) on the HTML e-form.

CGI scripts can be written in any language (NCSA, n.d. a). It is beyond the scope of this research to discuss specific implementations of CGI scripts. Therefore the general functionality and purpose of CGI scripts are discussed.

If the 'POST' method is used to submit e-form data, the CGI script reads the data from the standard input device as defined to the operating system. The length of the data is passed on to the CGI script through the operating system environment variable 'content_length'. If the 'GET' method is used to submit e-form data, the server passes the e-form information to the CGI script through the 'query_string' operating system environment variable (Marshall, 1996b).

The CGI script receives the data and translates it from an HTML format to one compatible with the ordering program. This format may even be EDI compliant. It forwards the translated data to the ordering program which may process the order. The ordering program then returns the processing results to the CGI script. The CGI script dynamically creates a standard HTML document with the results of the processing. The server is informed of the newly created web page through the standard output device. A command is sent to the web server to respond to the client with a status code 201 (Marshall, 1996b). The server returns the new web page created to the client through a 201 status code meaning that a new resource, i.e. a web page, has been created (Fielding, et al., 1997:55). The location response header field contains the address of the newly created web page - refer to section 6.5 in this chapter for more detail on location response header fields (Fielding, et al., 1997:51).

The order processing itself is a program external to the web server software. This is the reason CGI scripts are required to translate e-form data. One of the concerns noted with HTML forms is that they do not support client-side range checking for data values (Grobe, 1996). Therefore range checking must be performed by the ordering program.

CGI scripts have been introduced and order receipt, processing and response generation have been briefly discussed. This completes the final leg of the Internet ordering system.

2.9 Conclusion

The stated objectives of the chapter were to introduce the technological concepts used in an Internet ordering system, and to give a basic knowledge of the workings of such a system. This was achieved through:

- An explanation of client browser technology and the use of HTML forms;
- A discussion of the communication process, with HTTP and TCP/IP being introduced; and
- An overview of order translation, processing and response to the client, through discussion of CGI scripts.

The next step is to consider a risk model which may be applied to the technology, so that the risks in an Internet ordering system are identified. Chapter 3 will consider a risk model which may be appropriately applied to the technology discussed in this chapter.

Chapter 3

3. Risk Model

3.1 Introduction

The objective of this chapter is to select a risk model which may be suitably applied to the technology described in the previous chapter. Watne and Turney (1990:14) have recognised that the sophistication of computer systems has reached a point where auditors can no longer audit around the computer and they are forced to treat the computer as the target of an audit. In order to do this, the auditor must be aware of risks in the computer system being audited.

It is clear from Chapter 2 that the use of Internet ordering systems involves external networks. This diminishes the viability of relying on transaction audit trails and increases the reliance placed on data communications and involvement of third parties (SAICA, 1995:9). Thus the risk model selected must be able to identify and address the various components involved in an Internet ordering system.

3.2 Definition of risk

For the purposes of this short dissertation, the definition of risk used will be based on control objectives identified for computerised systems. These control objectives are presented in table 3.1.

Table 3.1: Control objectives (Watne and Turney, 1990:308-309)

Risk	*Description*
Transaction authorisation (1)	Transactions should be properly approved and authorised. Proper authorisation applies to both input transactions and transactions that are generated automatically by the computer during processing.
Errors are detected, corrected, and resubmitted (2)	Transactions that are in error should be detected. The erroneous data should be corrected and the transactions resubmitted for normal input and processing.

Table 3.1: Control objectives (Continued)

Risk	*Description*
Security of data and records (3)	Data and records should be secure against loss or destruction. Security should cover source documents, secondary file storage, and printed or microfilmed output.
Distribution of output (4)	Output should be distributed only to authorised individuals or departments.
Accuracy (5)	Input, processing and output should be accurate. Erroneous data, incorrect processing, and processing of wrong records or files should be avoided.
Uniqueness (6)	Transactions should not be entered into the computer or processed more than once.
Reasonableness (7)	Processing results should fall within a range of normal system results.

The South African Institute of Chartered Accountants recognises control risk as the risk that the internal control system may not prevent, detect or correct material misstatements (SAICA, 1996:1). Thus, for the purposes of this short dissertation, risk shall be defined as the risk that transactions processed through an Internet ordering system may not meet any one, or a combination of, the control objectives presented in table 3.1.

3.3 Risk Model

Having defined risk, an approach needs to be selected for identifying risks in an Internet ordering system. Due to the complex nature of such a system, the Path Context Model developed by Boshoff (1990) will be used as an approach to identifying risks in an Internet ordering system.

The Path Context Model was developed by Boshoff to address computer security in a significant number of situations (Boshoff, 1990:13-14). The model is based on

the concepts of a 'baggage collection vehicle', a security profile and a 'validator' (Boshoff, 1990:8). The 'baggage collection vehicle' is described as the mechanism of collecting end user and other information which needs to be transported across system boundaries in order to achieve the objectives of computer security (Boshoff, 1990:8). The security profile contains security rules or restrictions that need to be enforced (Boshoff, 1990:8). The 'validator' compares the 'baggage' to the 'security profile' and "provides the True or False condition for allowing access to secured objects" (Boshoff, 1990:8). Boshoff (1990:32-38) goes on to illustrate how a software or hardware component within a complex computer environment uses a 'validator' to compare the 'baggage' against the 'security profile' of that component. Based on the results of the validation test, access to the subsequent component within the access path may be allowed or disallowed (Boshoff, 1990:32-38). Boshoff's findings were that the more complex the environment was, the more effective the model was in addressing computer security (Boshoff, 1990:14).

The Path Context model will be applied to the various components of the Internet ordering system introduced in Chapter 2. This model will be presented in the form of a matrix, mapping control objectives against each identified component and the relevant sub-components in the Internet ordering system. Where a control objective has not been met, by default a risk exists for the control objective concerned.

3.4 Conclusion

The objective of this chapter was to identify a risk model which may be applied to Internet ordering systems. Due to the sophisticated nature of the system, it was noted that the computer auditor is compelled to audit through the computer. Risk in this type of environment was defined in the context of control objectives for computer systems, as identified by Watne and Turney (1990:308-309). Finally, the Access and Path Context Models developed by Boshoff (1985, 1990) were selected for application to identify the risks involved in these systems.

Chapter 2 discussed in detail the various components of Internet ordering systems. The functionality of these components was also described. Chapter 3 has defined risk and has identified a risk model which may be applied to the technology described in Chapter 2. Thus the focus of Chapter 4 will be to apply the risk model identified in Chapter 3 to the technology of Internet ordering systems described in Chapter 2.

Chapter 4

4. Application of Risk Model to Internet Ordering System Technology

4.1 Introduction

The objective of this chapter is to apply the risk model selected in Chapter 3 to the Internet ordering system, as presented in Chapter 2, and thereby identify risks in an Internet ordering system. Each of the components of this system will be arranged in a matrix and the control objectives will be mapped against the components. Where a particular control objective has not been met, by default a risk exists for that objective.

The risk matrix application to each layer within the Internet ordering system will be presented in the same order that the Internet ordering system components were introduced in Chapter 2.

4.2 Risk Matrices

For the purposes of the matrices, the following abbreviations apply:

- 1: Transactions are properly approved and authorised;
- 2: Errors are detected, corrected, and resubmitted;
- 3: Data and records are secure against loss or destruction;
- 4: Output is distributed only to authorised individuals or departments;
- 5: Input, processing and output are accurate;
- 6: Transactions are not entered into the computer or processed more than once; and
- 7: Processing results fall within a range of normal system results.

Where the matrix has been marked with a '√', this implies that the relevant control objective has been met for the component. Where the matrix is unmarked, the control objective has not been met and there is a risk for the relevant control

objective. The column marked "S" in the matrices reflects the relevant section in this short dissertation where the component is discussed. This column has been included in the control matrices to assist in understanding the reasons for the risk conclusions under the control objectives.

4.2.1 Client Browser and HTML formatted E-form

Table 4.1 maps the components of the client browser and HTML formatted e-form introduced in Chapter 2, against the control objectives for a secure system, introduced in Chapter 3.

Table 4.1: HTML Risk Matrix

Control Objective:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**S**
Component								
HTML								
Plain text								2.5.1
Multi-line entry								2.5.1
Option menus					√	√	√	2.5.2
Scrolled lists					√	√	√	2.5.2
Radio buttons					√	√	√	2.5.3
Check boxes					√		√	2.5.3
Password entry	√		√	√				2.5.4
Reset								2.5.5
Submit					√	√		2.5.5

Table 4.1: HTML Risk Matrix (Continued)

Control Objective:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**S**
Component								
HTML (Continued)								
'ACTION' parameter of '<FORM>' tag					√	√		2.5

The Gartner Group (1997a:1) recognises that HTML e-form technology lacks client-side logic for calculations and error checking, and on the server side it has insufficient compensation for calculations and error checking. Thus as a rule, with HTML, the components do not address control concerns and, consequently, result in risks in all risk categories. The exceptions, and reasons therefor, will now be discussed.

Option menus and scrolled lists provide the user with a list of choices which may be made. This reduces the risk of inaccurate choices being made. However, the user may select an option which he or she is not authorised to, or the user may avoid making a choice, leaving the field incomplete. This directly impacts on the risk of unauthorised choices or non-unique choices being made.

Radio buttons require that an option, and only one option, is selected. This mitigates the risk of incomplete data being sent. Furthermore, as the choices are predetermined, the risk of an inaccurate choice being selected or transmitted is mitigated. The risk factors with check boxes are similar to those with radio buttons, except that because none of the options, one of the options or more than one option may be selected, there is a risk that duplicates or omissions may occur.

The password entry input type provides a mechanism for the client to provide a password which may be verified on the server side to authenticate the user. This may potentially mitigate the risk of unauthorised transactions being processed.

The submit button initiates the transfer of the e-form data, thus contributing to the accurate and complete transfer of data from the client to the server. The 'ACTION' parameter of the '<FORM>' tag identifies the CGI script which will process the e-form data. Therefore it contributes positively to the accuracy and uniqueness.

4.2.2 HTTP

HTTP messages were introduced in Chapter 2. Two types of messages were noted, namely request and response message. HTTP messages were noted to comprise 'start lines', header fields, and response codes. These components are mapped against control objectives for secure computer environments in Table 4.2.

Table 4.2: HTTP Risk Matrix

Control Objective:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**S**
Component								
HTTP Request Message Start Line								
'METHOD'					√			2.6.1
Web page requested		√			√			2.6.1
HTTP Version		√			√			2.6.1
HTTP Request Header Fields								
Accept-Charset		√			√	√	√	2.6.2
Accept-Encoding		√			√	√	√	2.6.2
Accept-Language		√			√	√	√	2.6.2
Accept-Authorisation	√		√	√				2.6.2

Table 4.2: HTTP Risk Matrix (Continued)

Control Objective:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**S**
Component								
HTTP Request Header Fields (Continued)								
From	√		√	√				2.6.2
Proxy-Authorisation	√		√	√				2.6.2
Byte Ranges		√			√	√	√	2.6.2
Range Retrieval Request		√			√	√	√	2.6.2
User Agent	√		√	√				2.6.2
HTTP Response Message Status Line								
'PROTOCOL VERSION'		√			√	√	√	2.6.3
Status Code 100: Continue		√			√			2.6.4
Status Code 200: OK		√			√	√	√	2.6.4
Status Code 201: Created		√			√	√	√	2.6.4
Status Code 202: Accepted		√			√			2.6.4
Status Code 205: Reset Content		√			√	√	√	2.6.4
Status Code 305: Use Proxy	√		√	√				2.6.4
Status Code 400: Bad Request		√			√	√	√	2.6.4
Status Code 401: Unauthorised	√		√	√				2.6.4

Table 4.2: HTTP Risk Matrix (Continued)

Control Objective:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**S**
Component								
HTTP Response Message Status Line (Continued)								
Status Code 403: Forbidden	√		√	√				2.6.4
Status Code 407: Proxy Authentication Required	√		√	√				2.6.4
Status Code 408: Request Timeout						√		2.6.4
Status Code 409: Conflict		√			√	√		2.6.4
Status Code 411: Length Required		√			√	√		2.6.4
Status Code 414: Request-URI Too Long		√			√	√	√	2.6.4
Status Code 500: Internal Server Error		√			√	√	√	2.6.4
Status Code 501: Not Implemented		√			√	√	√	2.6.4
Status Code 502: Bad Gateway		√			√	√	√	2.6.4
Status Code 503: Service Unavailable						√		2.6.4
Status Code 504: Gateway Timeout						√		2.6.4
Status Code 505: HTTP Version Not Supported		√			√	√	√	2.6.4
HTTP Response Header Fields								
Location		√			√	√	√	2.6.5
Proxy-Authenticate	√		√	√				2.6.5

Table 4.2: HTTP Risk Matrix (Continued)

Control Objective:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**S**
Component								
HTTP Response Header Fields (Continued)								
Public		√			√	√	√	2.6.5
Retry-After						√		2.6.5
Warning		√			√			2.6.5
WWW-Authenticate	√		√	√				2.6.5

The reasons for the control objective conclusions are not discussed in detail here as these are clear from the description of the components as detailed in the column headed "S".

4.2.3 TCP/IP

Chapter 2 notes that both TCP and IP comprise of header fields which may impact on risk. Table 4.3 evaluates the relevant TCP and IP header fields against the control objectives identified in Chapter 3.

Table 4.3: TCP/IP Risk Matrix

Control Objective:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**S**
Component								
TCP Header Fields								
Source port	√		√	√				2.7.1
Destination port	√		√	√				2.7.1
Sequence number					√			2.7.1
Acknowledgement number					√			2.7.1
Data offset					√	√	√	2.7.1
Flags: ACK		√			√	√	√	2.7.1
Flags: PSH		√			√	√	√	2.7.1
Flags: RST		√			√	√	√	2.7.1
Flags: SYN					√	√		2.7.1
Flags: FIN					√	√		2.7.1
Window size						√		2.7.1
Checksum		√			√	√	√	2.7.1
Options					√	√	√	2.7.1
IP Header Fields								
Version number		√			√		√	2.7.3
Length						√		2.7.3

Table 4.3: TCP/IP Risk Matrix (Continued)

Control Objective:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**S**
Component								
IP Header Fields (Continued)								
Type of service		√			√	√	√	2.7.3
Packet length						√		2.7.3
Identification					√	√		2.7.3
Flags - More fragments					√	√		2.7.3
Flags - Don't fragment					√	√		2.7.3
Fragment offset					√	√		2.7.3
Time to live					√	√		2.7.3
Transport protocol		√			√	√		2.7.3
Header checksum		√			√	√	√	2.7.3
Source address	√		√	√				2.7.3
Destination address	√		√	√				2.7.3

Again, the reasons for the above control objective conclusions are clear from sections 7.1 and 7.3 of Chapter 2. It should be noted, though, that the control objective conclusions for IP relate to the IP packets and not the data contained therein. It was observed in section 7.3 of Chapter 2 that the only IP header field which considers the data included in the packet is the 'packet length' header field. The other IP header fields are only concerned with the IP header itself. Furthermore, it should also be noted that IP does not guarantee forwarding of packets to the next destination (Santifaller, 1994:19). Clearly this increases the risk

that packets may not be delivered to their destinations. However, TCP addresses this risk.

4.2.4 CGI Scripts and Supplier Ordering systems

It is not possible to present a detailed control objective model in a matrix format for CGI scripts or for supplier ordering systems. This is because they both are a form of an application program, and an appropriate application review would have to be performed for each unique CGI script and ordering system, to identify whether or not control objectives have been met and the relevant risks addressed in these systems. However, a simple matrix considering the effect of operating system environment variables is presented in Table 4.4

Table 4.4: CGI Script Risk Matrix

Control Objective:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**S**
Component								
Operating System Environment Variables								
Content Length						√		2.8
Query String	√	√	√	√	√	√	√	2.8

4.3 Conclusion

The aim of this chapter was to apply a control objective model to an Internet ordering system and thereby identify risks involved in such a system. The technology infrastructure described in Chapter 2 was mapped against the control objective model selected in Chapter 3, with the result that the risks in such a system were identified.

Having completed this research, it is necessary to look at the process which was followed and to identify whether or not the objectives set out in Chapter 1 were met. Furthermore, it is also necessary to consider other fields of research which have been identified in the course of the present study. These issues will be addressed in Chapter 5.

Chapter 5

5. Conclusion

5.1 Introduction

The aim of this chapter is to assess whether the objectives set out in Chapter 1 were met and to identify other areas requiring further research, within the framework of this short dissertation and for areas specifically excluded from the scope of this short dissertation.

5.2 Applicability of the Risk Model

The stated objective of this research was to develop a risk model which can be applied to Internet ordering systems. A model was developed in Chapter 4, with consideration given to each layer of components within the process. The model may be applied generically to any Internet ordering system, regardless of the platforms on which the client or the server are operating. Thus the model may have applicability in situations where Internet ordering systems are implemented through HTML e-forms and where such forms are communicated from the supplier to the customer (and vice versa) through the Internet. It should be noted that the supplier and the client may have little influence on the development of web servers and browsers respectively. This is due to the fact that such tools are commercially available. As was illustrated in Chapter 2, the web browser and the server interact closely with the HTTP, which, in turn interacts with TCP/IP and the physical network connections. Thus one may believe that customers and suppliers have little control over the risks in HTTP and TCP/IP. However, suppliers may influence the design of the e-form (through the use of HTML to design such forms), the functionality of CGI scripts and the interaction between CGI scripts and the ordering application. Therefore, it would appear that risk and control issues may be addressed on the supplier side through development of appropriate e-forms, CGI scripts and interface controls between the ordering system and CGI scripts. Therefore, auditors may need to focus risk assessment and control evaluation efforts on the supplier's server side.

Despite this, ignoring platform and application-specific issues has certain implications. If one is to consider application of the Access Path model and the evaluation of risk in a computer environment, the model is required to be applied to all aspects of a system from the point of origin of a transaction (at the client) through to the point where the data is finally stored on the system (at the server). Therefore platform and application-specific issues cannot be ignored.

The result of this is that the risk model developed needs to be used with risk models developed for other aspects of computerised ordering systems. The model only forms part of the risk evaluation tools and techniques which are available to the computer auditor to use.

5.3 Other Fields Opened for Research

The discussion in section 2 of this chapter raises the issue of areas which need to be investigated further. It was noted in Chapter 2 that Internet ordering systems may also be implemented through encoding EDI messages in a MIME format, and using the Internet to transport such messages, rather than using a Value Added Network or a dedicated line. Discussion of this implementation of an Internet ordering system was beyond the scope of this research.

An area briefly reviewed in this short dissertation which needs further research is the development of security features available within specific CGI script implementations which interface with ordering systems. This was not addressed in this short dissertation due to the different system-specific CGI implementations that are available.

The application of the Access Path model to the application system, the database management system and the operating system in specific implementations of Internet ordering systems was not addressed in this short dissertation. It was beyond the scope of the research.

Despite this, the risk model developed in this short dissertation will assist auditors in understanding the framework in which Internet ordering systems are implemented and it will provide computer auditors with a basis upon which to evaluate risks in an Internet ordering system. The achievement of these goals will assist auditors in designing an effective and efficient audit approach for businesses where Internet ordering systems are the source of material sales transactions.

In conclusion, growth of electronic commerce on the Internet is being hampered by two issues (The Gartner Group, 1997b:1), namely:

- whether or not products purchased over the Internet should be taxed; and
- whether or not export of sophisticated encryption software from the United States should be restricted.

However, The Gartner Group believes that these issues are close to resolution (The Gartner Group, 1997b:1) for two reasons. Firstly, the US government had proposed, in July 1997, that all electronically delivered Internet goods and services be duty free. In the same month, European governments agreed to adopt the US stance on Internet delivered goods and services (The Gartner Group, 1997b:1). Secondly, several patents relating to encryption techniques are expiring in the near future (The Gartner Group, 1997b:2-3). Therefore The Gartner Group (1997b:1-3) expects commerce over the Internet to get a boost.

Bibliography

1. Boshoff, WH 1990: "A Path Context Model for Computer Security Phenomena in Potentially Non-Secure Environments", dissertation in fulfilment of a Doctors Degree in Natural Sciences; Rand Afrikaans University.

2. Boshoff, WH 1985: "The Interface between Applications and Integrity Controls in Modern Computer Systems", dissertation in fulfilment of a Masters Degree in Economic Sciences; Rand Afrikaans University.

3. Carey, P & Ambrosia A 1995: The Internet Illustrated; Cambridge, MA: International Thomson Publishing, Inc.

4. Derfler, Jr. & Freed L 1993: How Networks Work; Emeryville, CA: Ziff-Davis Press.

5. Emmelhainz, MA 1990: Electronic Data Interchange, A Total Management Guide; New York: Van Nostrand Reinhold.

6. Fielding, R; Irvine, UC; Gettys, J; Mogul, J; DEC; Frystyk, H; Berners-Lee, T & MIT/LCS 1997. *Hypertext Transfer Protocol – HTTP /1.1*. [Online]. Available URL address: http://www.ics.uci.edu/pub/ietf/http/rfc2068.txt.

7. Grobe, M 1996. *An Instantaneous Introduction to CGI Scripts and HTML Forms*. [Online]. Available URL address: http://kuhttp.cc.ukans.edu/info/forms/forms-intro.html.

8. Marshall, J 1996a. *CGI Made Really Easy*. [Online]. Available URL address: http://www.jmarshall.com/easy/cgi.

9. Marshall, J 1996b. *Footnotes for CGI Made Really Easy*. [Online]. Available URL address: http://www.jmarshall.com/easy/cgi_footnotes.

10. NCSA, University of Illinois at Urbana Champaign 1996. *A Beginner's Guide to HTML*. [Online]. Available URL address: http://www.ncsa.uiuc.edu/General/Internet/WWW/HTMLPrimerAll.html.

11. NCSA, University of Illinois at Urbana Champaign n.d. a. *Common Gateway Interface*. [Online]. Available URL address: http://hoohoo.ncsa.uiuc.edu/cgi/intro.html.

12. NCSA, University of Illinois at Urbana Champaign n.d. b. *Decoding Forms with CGI*. [Online]. Available URL address: http://hoohoo.ncsa.uiuc.edu/cgi/forms.html.

13. NCSA, University of Illinois at Urbana Champaign n.d. c. *The Common Gateway Interface*. [Online]. Available URL address: http://hoohoo.ncsa.uiuc.edu/cgi/primer.html.

14. Puttick, G & van Esch S 1992: The Principles and Practice of Auditing; revised sixth edition. Cape Town: Juta & Co Ltd.

15. Ragget, D 1997. *HTML 3.2 Reference Specification*. [Online]. Available URL address: http://www.astro.ulg.ac.be/html-rec.32/rec-html.html.

16. SAICA 1995: Paperless Business Transactions; Kengray: The Natal Witness Printing Publishing Company (Pty) Ltd.

17. SAICA 1996: SAAS 400 - Risk Assessments and Internal Control.

18. Santifaller, M 1994: TCP/IP and ONC/NFS, Internetworking in a UNIX Environment; second edition. Cambridge, United Kingdom: Addison-Wesley.

19. The Garnter Group 1997a: Enterprise Applications and the Internet: Key Questions; Research Note. [Online]

20. The Gartner Group 1997b: Globalizing Commerce Over the Internet; Point-To-Point. [Online]

21. The Gartner Group 1996a: The Electronic Workplace: Fulfilling the Promise, Part 1; Industry Trends & Directions (ITD).

22. The Garnter Group 1996b: Integrated E-Form Management Strategies; Office Information Systems (OIS).

23. The Garnter Group 1996c: The Physical Structure of the Internet; Research Note.

24. University of Toronto n.d. *An Introduction to the Common Gateway Interface*. [Online]. Available URL address: http://www.utoronto.ca/webdocs/CGI/cgi1.html.

25. Watne, DA & Turney PBB 1990: Auditing EDP Systems; second edition. New Jersey: Prentice-Hall Inc.