Global Information Assurance Certification Paper

Technical security audit of a customer support web application portal: the independent auditor perspective

Marcel M. B. Carlsson
2003 Nov 24

Abstract

This paper details a technical audit of a customer support web application portal that is publicly available on the internet. The scope of the audit is limited to security risks relevant to the web application and the web application server. The audit findings can be regarded as typical for businesses and organisations that lack the appropriate security skills and suitable resources to address web application security requirements and secure development, implementation and operation. The audit will uncover a web application riddled with many of the most common web application security vulnerabilities and lack of adequate general security management. The basic audit steps of risk assessment, check list development, auditing and management reporting will be covered in this paper.

Introduction

This audit has been done from an independent auditor's point of view. The definition of the term "independent auditor", used in this papers, refers to the definition provided in the SANS Network and Systems Auditing Track training course. The term refers to an outside third-party brought in to review a system. The SANS definition also makes the assumption that independent auditors have no pre-existing knowledge of the system (other than what may be provided to them in the course of conducting the audit) and in this case also do not have administrative or root-level control over the system being audited.

Research in Audit, Measurement Practice and Control

System Identification

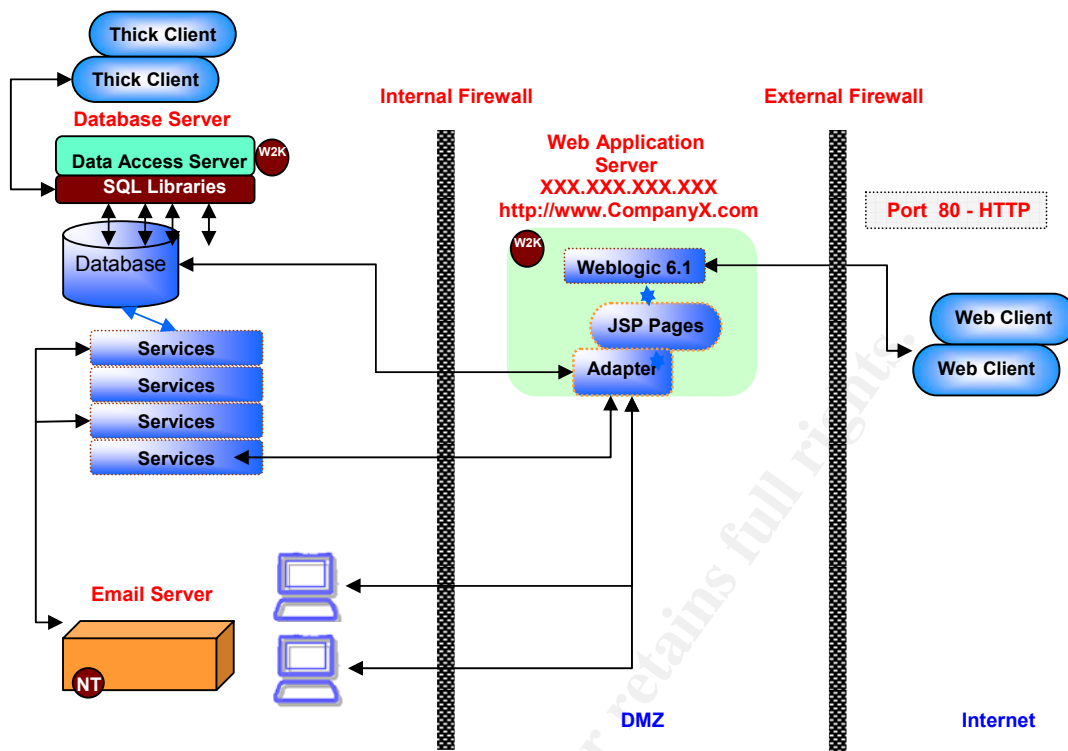### General

The audited system is one of the CompanyX internet facing public web application portals. The portal is based on a popular Customer Relationship Management (CRM) software package. The package has been customised by the CompanyX web application development team to meet the specific needs of the business.

The portal provides business functionality that enable CompanyX's clients to conduct problem management and release management through a web browser interface. In addition, the portal also provides authentication services, a role-based access control mechanism and a navigational framework.

The web application development team is based at the CompanyX development site where development and testing activities take place. The web application production environment is hosted in CompanyX's data centre, which is physically separate from the development site. The web application team remotely administers the web application and the web application server. The CompanyX staff in the data centre is responsible for delivering operations and support of the portal production infrastructure, to the web application team.

The web application technical architecture is a basic three-tier design. The web application server is housed between the external and internal firewall in the demilitarised zone (DMZ). The web application is implemented with java server pages (JSP) technology. The web application server is a Weblogic 6.1 server running on a Windows 2000 platform. The high level technical architecture of the web application is shown in the diagram below, as provided to the auditor by the application development team.

Thick Client

Thick Client

Database Server

Data Access Server | **W2K**

SQL Libraries

Database

Services

Services

Services

Services

Email Server

NT

Internal Firewall

External Firewall

Web Application Server
XXX.XXX.XXX.XXX
http://www.CompanyX.com

W2K

Weblogic 6.1

JSP Pages

Adapter

Port 80 - HTTP

Web Client

Web Client

DMZ

Internet

Scope

This is a technical web application security audit and the scope is limited to publicly available assets:

- Web Application Security
- Web Application Server Security

However, the web application data flow passes through the external firewall and this part of the infrastructure will be examined with a scanning tool.

Although the scope is explicitly focused on web application security and web application server security, there may be minor overlaps into other areas of security. These overlaps will be included in the report if they add value to this audit but overlaps will be considered on a high level with minimal supporting details. Examples of these overlaps include operational security related to the web application and the web application server and relevant security management issues.

Non-public assets and other areas of security not related to web application security and web application server security are not within scope for this audit:

- Information Technology Security
  - o Firewalls
    - Internal
    - External (except scanning open ports)
  - o Back-end Servers
    - Database Server
    - Email Server

- o Thick Clients
- o Remote Administration Workstations
- o Platform security (operating systems)
- Physical Security
- Business Operations Security (except related to the web application and the web application server)
- Business Process Security
- Security Management (except related to the web application and the web application server)

Risk Assessment

Basic Methodology

There are many security risk assessment methodologies out there in the business security field to choose from. From this auditor's own experience it is sometimes possible to get bogged down in detailed analysis work and lose sight of the big picture. In order to keep things simple and concise for this audit, we will apply the simple five step security risk management methodology that Bruce Schneier discusses in his book "Beyond Fear: Thinking Sensibly About Security in an Uncertain World". Schneier's methodology is based on answering the following five questions:

1. What business critical assets need to be protected?
2. What are the risks to these assets?
3. What are the security solutions?
4. What new risks do these security solutions create?
5. What are the trade-offs among key stakeholders in the system considered?

Business Critical Assets

The critical business assets that need to be protected is the data flow processed in the web application and on the web application server. This flow contains personal information such names and addresses. These are of a private nature and are subject privacy laws. Other data flows include information that customers input to the system to manage information technology related problems and software releases. This audit will not cover the security of the stored data in the back-end database itself.

The web application hardware and software assets are also critical business assets since without these there would be no web application service available to the customers. In our case, within scope is the web application running on the web server, the web application server software and the web server hardware. The underlying operating system is out of scope since the auditor will not have remote access to the registry on the Windows 2000 box.

Finally, the company also has its reputational asset to protect. Ending up on the front page of the national newspaper with a negative headline would not make executive management, shareholders and business partners happy, not

to mention potential loss of customer confidence and the prospect of dipping company stock prices.

Risks

Let us first look at the risks to the informational assets. In his book "Fighting Computer Crime: A New Framework for Protecting Information", Donn B. Parker has extended the standard confidentiality, integrity and availability triad and made it more comprehensive to include (please refer to Parker's book for more details):

- Availability and Utility
- Integrity and Authenticity
- Confidentiality and Possession

Applying Parker's concepts and considering potential risks to the critical business assets CompanyX should be protecting, we come up with the threats listed in the risk assessment summary table below.

We will use a simple relative three-step scale and assign likelihoods of low, medium and high to each threat as specified in the NIST Risk Management SP800-30 publication. These are relative levels and serve the purpose to identify risk areas which should be taken more serious than others.

Likelihood Level	Likelihood Definition
High	The threat-source is highly motivated and sufficiently capable, and controls to prevent the vulnerability from being exercised are ineffective.
Medium	The threat-source is motivated and capable, but controls are in place that may impede successful exercise of the vulnerability.
Low	The threat-source lacks motivation or capability, or controls are in place to prevent, or at least significantly impede, the vulnerability from being exercised.

Similarly for impact, we use a rating of low, medium and high. Again, these are qualitative and subjective judgements that enable us to get a feel for which impacts are more serious than others. These are also taken from the NIST Risk Management SP800-30 publication.

Magnitude of Impact	Impact Definition
High	Exercise of the vulnerability may result in the highly costly loss of major tangible assets or resources. May significantly violate, harm, or impede CompanyX's business, reputation or interest. May result in human death or serious injury.
Medium	Exercise of the vulnerability may result in the costly loss of tangible assets or resources. May violate, harm, or impede CompanyX's business, reputation or interest. May result in human injury.
Low	Exercise of the vulnerability may result in the loss of some tangible assets or resources. May noticeably affect CompanyX's business, reputation or interest.

The risk levels are simply calculated by using the matrix below based on the corresponding levels of likelihoods and impacts. These are taken from the NIST Risk Management SP800-30 publication. For this exercise we do not have any hard quantitative data, which is required to perform an analysis backed up by numbers. The nature of the available information is purely qualitative and it makes little sense to assign numbers or even use a more detailed scale of measures since we are only assigning relative levels to likelihood, impact and risk to get a high level understanding of the relative risk levels.

		Impact		
		L	**M**	**H**
Likelihood	**L**	L	L	M
	M	L	M	H
	H	M	M	H

Below are the definitions what these risk levels mean based on the NIST Risk Management SP800-30 publication.

Risk Level	Description
High	If an observation or finding is evaluated as a high risk, there is a strong need for corrective measures. An existing system may continue to operate, but a corrective action plan must be put in place as soon as possible.
Medium	Medium If an observation is rated as medium risk, corrective actions are needed and a plan must be developed to incorporate these actions within a reasonable period of time.
Low	If an observation is described as low risk, the business must determine whether corrective actions are still required or decide to accept the risk.

Risk Assessment Summary

The table below summarizes the risk assessment findings, based on the information provided to the auditor by employees at CompanyX, and is a result of following the five step methodology discussed earlier in this section.

Ref	Assets	Threats	Likelihood (H/M/L)	Impact (H/M/L)	Risk (H/M/L)
WSS1	Web Server Software	No defined security requirements	H	H	H
WSS2		Not hardened	H	H	H
WSS3		Not patched	H	H	H
WA1	Web Application	No defined security requirements	H	H	H
WA2		Weak or no authentication	M	H	H
WA3		Weak or no access control	M	H	H
WA4		Information leakage	M	M	M

Ref	Assets	Threats	Likelihood (H/M/L)	Impact (H/M/L)	Risk (H/M/L)
WA5		Improper use of encryption	M	M	M
WA6		Insecure session management	M	H	H
WA7		Poor input validation	H	H	H
WA8		Poor logging	M	M	M
WA9		Poor change control	L	M	M
WA10		No secure coding practice	H	H	H
WA11		Lack of security testing	H	H	H
WA12		Not patched	H	H	H
INF1	Information	Deleted on purpose	L	H	M
INF2		Deleted by mistake	M	H	H
INF3		Modified on purpose	L	M	M
INF4		Modified by mistake	L	M	M
INF5		Disclosed on purpose	L	H	H
INF6		Disclosed by mistake	L	H	H
INF7		Unauthorised copying	H	H	H

Ref	Assets	Threats	Likelihood (H/M/L)	Impact (H/M/L)	Risk (H/M/L)
REP1	Reputation	Web site hacked and defaced	L	H	M
REP2		Denial of service attack renders service out of business	L	H	M
REP3		Insider compromises data and sells to competitor or criminal	H	H	H

Possible Solutions

Considering the risk areas and risk levels we then proceed to propose suitable controls that will mitigate the risks. Controls can be technical, procedural or managerial in nature.

Additional Risk

The concept of absolute zero level risk does not exist in real life. There will always be some kind of risk level present. Even after implementing suitable controls to bring down the risk level. Remember that we are interested in managing business security risk and ensuring that we are in control, i.e. the risk levels should be at a level the business considers acceptable.

One thing to remember is that adding security controls may bring new risks. As Schneier points out, the goal is to make sure that the sum all risk levels due to newly introduced security controls must be less than the risk level that we started out with when we had no controls. Schneier also makes the point that if we do not do this step, we may not only instil a false sense of security but also spend time and effort on security controls that may be effectively useless.

Ref	Security Solution	New Risks Due To Proposed Security Solution
WSS1	Extract security requirements from business requirements. Conduct a security risk assessment. Follow security best practice.	-

Ref	Security Solution	New Risks Due To Proposed Security Solution
WSS2	Define process for hardening web application server software. Apply change control.	-
WSS3	Define process for keeping up to date regarding new patches, testing and installing patches. Apply change control.	Patches not tested or patches break production environment.
WA1	Extract security requirements from business requirements. Conduct a security risk assessment. Follow security best practice.	-
WA2	Define security requirement. Follow best practice. Implement appropriate authentication mechanism.	-
WA3	Define security requirement. Follow best practice. Implement appropriate access control mechanism.	-
WA4	Define process for cleaning up content that will be published on the web site. Follow best practice.	-
WA5	Define security requirement. Follow best practice. Implement appropriate encryption control mechanism.	Encryption keys not managed securely.
WA6	Implement a session management mechanism that is secure. Follow best practice.	-
WA7	Implement an input validation mechanism that is secure. Follow best practice.	-
WA8	Define security requirement. Follow best practice. Implement appropriate logging mechanism.	Audit logs not protected from unauthorised access, deletion or modification.
WA9	Implement a formal change	-

Ref	Security Solution	New Risks Due To Proposed Security Solution
	control process. Follow best practice.	
WA10	Educate developers regarding common security vulnerabilities. Follow best practice.	-
WA11	Implement a security testing into the system development life cycle at all stages. Follow best practice.	Vulnerabilities may not be put into the context of business requirements.
WA12	Define process for keeping up to date regarding new patches, testing and installing patches. Apply change control.	Patches not tested or patches break production environment.
INF1	Implement authentication control, access control, audit trails, backup and recovery process.	Backup media is not protected from unauthorised access.
INF2	Implement authentication control, access control, audit trails, backup and recovery process.	Backup media is not protected from unauthorised access.
INF3	Implement authentication control, access control, audit trails, backup and recovery process.	Backup media is not protected from unauthorised access.
INF4	Implement authentication control, access control, audit trails, backup and recovery process.	Backup media is not protected from unauthorised access.
INF5	Implement authentication, access control and encryption mechanism.	Encryption keys not managed securely.
INF6	Implement authentication, access control and encryption mechanism.	Encryption keys not managed securely.
INF7	Implement authentication, access control, audit trails and encryption mechanism.	Encryption keys not managed securely.

Ref	Security Solution	New Risks Due To Proposed Security Solution
REP1	Implement access control and integrity checking control.	-
REP2	Define security requirement and implement resilient solution accordingly.	-
REP3	Implement authentication, access control, audit trails and encryption mechanism.	Encryption keys not managed securely.

Trade-Offs

Implementing and maintaining security controls requires a financial budget and resources. These factors will have to be considered by the business when selecting security controls. In addition, as Schneier points out, running a business usually involves complex relationships between various stakeholders who all have different agendas and their own business reasons. Often these agendas and reasons are conflicting. The tricky part for the security professional is to strike a good balance between all these trade-offs and end up with adequate security levels.

For example, during application development the last thing the application development manager wants to hear is that he or she needs to add in an additional security control in the application to make the application more secure. Such a change is likely to in the short term negatively impact the application development project plan, resourcing and budget. However, in the long term it is less costly and more wisely to address security as early as possible in the system development cycle. The same is true for most other areas in the company, whether the security control is technical, procedural or managerial.

Trade-offs must thus be made to ensure that a sensible balance is struck between security and running a profitable business. A detailed discussion on trade-off exercise is out of scope for this paper but an excellent resource for a detailed example of such a process and an explanatory example on how conduct such an analysis can be found in the NIST Risk Management SP800-30 publication.

Current State of Practice

After conducting research in the area of web application security auditing, the auditor's impression is that this is an emerging area with much room for improved methodologies and formal standards. However, the emerging web application security auditing area is quickly maturing and there are a number

of good initiatives that provide a wealth of resources for auditors that need to get their hands dirty with web application security auditing.

The premium source is the Open Web Application Source Project (OWASP), which have produced a comprehensive guide to securing web applications. The guide is called the "OWASP Guide to Building Secure Web Applications". OWASP also hosts a website, which contains information and tools for web application security professionals. OWASP also have released the top ten web application vulnerability list that details the most common security problems with web applications. The list is called "OWASP Top Ten Most Common Web Application Vulnerabilities". The success and power of the OWASP initiative lies in its active members that proactively share and exchange information freely giving back to the security community. Since OWASP can be seen as the current open source authority on web application security, it will form much of the basis for this technical audit.

For cutting edge technology specific web application security information there is also the Black Hat Briefings site, which hosts a series of presentation slides and hour long presentations in digital video format to be freely downloaded. The URL of this site is http://www.blackhat.com. Click on the briefings link to go to the interesting content from past briefings. Many of the web application attack methodologies, vulnerabilities and tools discussed in the OWASP guide, were first presented and released at past Black Hat Briefings.

There are also a number of good books that cover interesting topics in web application security, "Hacking Web Applications Exposed", "Special Ops: Host and Network Security for Microsoft, UNIX, and Oracle" and "Web Hacking: Attacks and Defense". For problem solving style scenarios, which include web application security, there are the books "Hacker's Challenge", "Hacker's Challenge 2" and "How to own the box", which bring a fresh take on security literature and explore the attacker mentality.

There are also a number of interesting white papers that have been written about web application security by many of the authors and presenters mentioned in this section. Please go to their respective websites for the latest published white papers. One web page that has links to many of these papers is at the CGI Security website (http://www.cgisecurity.com/lib/).

David Rhoades' web application security auditing module in the SANS Audit Track course literature is excellent. This module gives a well-thought out overview of what to include in a web application security audit and discusses tools that may be used to do this. SANS also have a web application audit check list at their S.C.O.R.E website.

With regards to automated web application security auditing tools, a number of tools have limited automated web application security audit functionality built-in. These tools will check for known vulnerabilities and check for default installation weaknesses such as common demo passwords, files and applications that may be exploited. Examples of these tools include eEye Retina and Foundstone Foundscan, which both are used in this web

application security audit. However, to find security problems within the business logic of web applications, human intervention is usually required, since several low risk vulnerabilities may become high risk vulnerabilities when combined or exploited in a clever combination to compromise the web application. Please refer to Grossman's presentation "Challenges of Automated Web Application Scanning", given at Blackhat Briefings Federal 2003 for more information on this topic.

The resources mentioned above mainly deal with the technology side of web application security. However, security policies are equally, if not more, important since these can be regarded as the glue between the security requirements and the security implementation. However, the ISO17799 framework, at the time the paper is being written, lacks specifics regarding web application security. The Information Security Forum (ISF) and their "The Standard of Good Practice for Information Security" is another framework, which disappoints in the specific area of web application security auditing.

Audit Checklist Development

Unfortunately CompanyX, the owner of the target system for this audit, has not implemented any formal security policies or documented any security procedures. According to the CompanyX, security requirements were never explicitly addressed or defined when business requirements were agreed. This in itself is a major security concern. In light of this, this technical audit will be benchmarked against accepted web application security industry best practice.

Audit Item 1 - Validated Parameters

Reference	OWASP Guide (page 45-48), Web Application Checklist (item seven), 7.3. Auditing Web-based Applications (page 204)
Control Objective	To ensure that information from web requests is validated before being used by a web application.
Risk Area	Attackers can use these flaws to attack back-end components through a web application. WA7. WA10. REP1.
Compliance	The web application must block all input unless it is explicitly allowed. In other words, deny all input except known "good" input that is expected by the application. This must apply to any parameter that is passed to the web application.
Test Method	1. Log onto the application as a typical user. 2. Locate areas in the web application where the user provides input to the web application. 3. Inject data that should be blocked by the web application: < > ; ! * /.. NULL 4. Examine if data is validated by the web application. 5. If parameters are not validated then this means non-compliance.
Test Type	Objective

Audit Item 2 - Secure Access Control

Reference	OWASP Guide (page 27-28)
Control Objective	To ensure that restrictions on what authenticated users are allowed to do is properly enforced. WA3. WA10. REP1.
Risk Area	Attackers can exploit these flaws to access other users'

	accounts, view sensitive files, or use unauthorized functions.
Compliance	The access control mechanism for the web application must be implemented according to the design and meet security requirements.
Test Method	1. Log on to the application as a specific user with specific access rights. 2. Examine if it is possible to access areas of the applications, which are not allowed by design. 3. If access can be gained to unauthorised areas then this means non-compliance.
Test Type	Objective

Audit Item 3 - Secure Account Management

Reference	OWASP Guide (page 16-19), 7.3. Auditing Web-based Applications (page 179)
Control Objective	To ensure that account credentials are properly protected.
Risk Area	Attackers may compromise passwords, keys, session cookies, or other tokens that can defeat authentication restrictions and assume other users' identities. WA2. WA10. NF5. INF6. REP1.
Compliance	The account management for the web application adequately protect the information and ensure that is not accessed by unauthorised users.
Test Method	1. Examine the web authentication mechanism. 2. Identify a password brute-forcing tool. 3. Obtain suitable username and password lists to be tested. 4. Configure the tool and setup a password brute-forcing session. 5. Run the tool and examine if the password mechanism can be subverted. 6. If the application allows the brute-forcing tool to run interrupted without slowing down or locking out the attack, then this means non-compliance since it is only a matter of time until the password will be cracked.
Test Type	Objective

Audit Item 4 - Secure Session Management

Reference	OWASP Guide (page 22-26), 7.3. Auditing Web-based Applications (page 156)
Control Objective	To ensure that sessions are securely managed.
Risk Area	Attackers may predict, replay or brute force sessions and hi-jack other users' sessions and gain access to information and services which they do not have authorisation for. WA6. WA10. INF5. INF6. REP1.
Compliance	Session management must be securely implemented to prevent sessions IDs from being predicted, brute forced or compromised in any way.
Test Method	1. Use a local web application proxy tool, log onto the application as a typical user. 2. Sample a series of web application session IDs. 3. Examine the session IDs to make sure that they can not be guessed or brute-forced. 4. If session IDs can be predicted then this means non-compliance.
Test Type	Objective

Audit Item 5 - Cross-Site Scripting (XSS) Controls

Reference	OWASP Guide (page 34-35), Web Application Checklist (item six), 7.3. Auditing Web-based Applications (page 191)
Control Objective	Ensure that the web application can not be used as a mechanism to transport an attack to an end user's browser.
Risk Area	A successful attack can disclose the end user's session token, attack the local machine, or spoof content to fool the user. WA7. WA10. INF5. REP1.
Compliance	The web application must not allow code to be injected and displayed back in the users' browsers.
Test Method	1. Use a local web application proxy tool, log onto the application as a typical user. 2. Locate areas in the web application where the user provides input to the web application. 3. Intercept and manipulate data.

	4. Examine if web application is vulnerable to cross site scripting attacks.
	5. If the web application accepts code snippets as input and pipes this back to the client side then this means non-compliance.
Test Type	Objective

Audit Item 6 - Buffer Overflow Controls

Reference	OWASP Top Ten (page 12)
Control Objective	Ensure that the web application components are not vulnerable to buffer overflow attacks.
Risk Area	Web application components in some languages that do not properly validate input can be crashed and, in some cases, used to take control of a process. These components can include CGI, libraries, drivers and web application server components. WA7. WA10. REP1.
Compliance	None of the web application components must be vulnerable to buffer overflow attacks.
Test Method	1. Use an automated web application scanning tool. 2. Search for any known buffer overflow weaknesses. 3. Perform a web application code review. 4. Use an automated buffer flow analysis tool to locate any potential buffer overflow vulnerabilities in the web application and its components. 5. If any buffer overflow weaknesses are found then means non-compliance.
Test Type	Objective

Audit Item 7 - Command Injection Controls

Reference	OWASP Top Ten (13-14), Web Application Checklist (item 8)
Control Objective	To ensure that the web application is not vulnerable to command injection attacks.
Risk Area	Web applications pass parameters when they access external

	systems or the local operating system. If an attacker can embed malicious commands in these parameters, the external system may execute those commands on behalf of the web application. WA7. WA10. REP1.
Compliance	None of the web application components must be vulnerable to code injection attacks.
Test Method	1. Update the automated web application scanning tool with the latest known vulnerability signatures. 2. Run the tool and to find any command injection weaknesses. 3. Use a local web application proxy tool and locate areas in the web application where the user provides input to the web application. 4. Intercept and manipulate data. 5. Examine if web application is vulnerable to code injection attacks. 6. If the web application is vulnerable to code injection or even leak any error messages that may aid an attacker then this means non-compliance.
Test Type	Objective

Audit Item 8 – Secure Error Handling

Reference	OWASP Top Ten (page 15-16), Web Application Checklist (item twenty-two)
Control Objective	To ensure that web application error conditions that occur during normal operation are handled properly.
Risk Area	If an attacker can cause errors to occur that the web application does not handle, they can gain detailed system information, deny service, cause security mechanisms to fail, or crash the server. WA4. WA10. INF5. INF6. REP1.
Compliance	The web application error messages must not leak any information that can aid attackers in compromising the security of web application.
Test Method	1. Use a local web application proxy tool, log onto the application as a typical user. 2. Locate areas in the web application where the user provides input to the web application.

	3. Intercept and manipulate data and force error messages.
	4. Examine if returned error messages do not leak any sensitive data that might aid an attacker.
	5. If sensitive data is contained in any error message then this means non-compliance.
Test Type	Objective

Audit Item 9 - Secure Use of Cryptography

Reference	OWASP Top Ten (page 17-18), Web Application Checklist (item twenty-six), 7.3. Auditing Web-based Applications (page 120)
Control Objective	To ensure that the web applications use cryptographic functions to protect information and credentials, where required. **Note!** This audit item is limited to the use of cryptographic controls within the web application. Cryptographic controls for data stored in the back-end data base is out of scope).
Risk Area	These cryptographic functions and the code to integrate them have proven difficult to code properly, frequently resulting in weak protection. WA5. WA10. INF5. INF6. REP1.
Compliance	The web application must have implemented cryptographic controls securely according to security requirements.
Test Method	1. Use a local web application proxy tool, log onto the application as a typical user.
	2. Locate areas in the web application where highly sensitive information is processed.
	3. Examine if cryptographic protocols have been implemented to protect the sensitive data.
	4. If sensitive data is not protected in by cryptographic controls then this means non-compliance.
Test Type	Subjective

Audit Item 10 – Secure Remote Administration

Reference	OWASP Top Ten (page 19-20)
Control Objective	To ensure that the web application allows administrators to securely remotely administer the web application.

Risk Area	If these administrative functions are not carefully protected, an attacker can gain full access to all aspects of a site. WA2. WA3. WA10. REP1.
Compliance	All web application remote administration interfaces must be locked down according to security requirements.
Test Method	1. Update the automated web application scanning tool with the latest known vulnerability signatures. 2. Run the tool and to locate any running remote administration service interfaces. 3. Examine if these interfaces have been securely implemented. 4. Launch a brute-force attack. 5. If the remote administration has not been locked down properly then this means non-compliance.
Test Type	Subjective

Audit Item 11 – Secure Web Application Configuration

Reference	OWASP Top Ten (page 21-22), Web Application Checklist (item eight)
Control Objective	To ensure that the web application configuration security standard security meets the defined security requirements and follows security best practice. WA2. WA3. WA10. REP1.
Risk Area	Web applications have many configuration options that affect security and are not secure out of the box.
Compliance	The web application must be configured securely to meet security requirements and the configuration must be under strict configuration management control.
Test Method	1. Update the automated web application scanning tool with the latest known vulnerability signatures. 2. Run the tool and probe for any known vulnerabilities due to misconfiguration. 3. If the any known vulnerabilities are detected then this means non-compliance.
Test Type	Objective

Audit Item 12 – Secure Web Server Configuration

Reference	OWASP Top Ten (page 21-22), Web Application Checklist (item eight)
Control Objective	To ensure that the web server configuration security standard security strong.
Risk Area	Web servers have many configuration options that affect security and are not secure out of the box. WSS2. WSS3. REP1.
Compliance	The web server must be configured securely to meet security requirements and the configuration must be under strict configuration management control.
Test Method	1. Update the automated web application scanning tool with the latest known vulnerability signatures. 2. Run the tool and probe for any known vulnerabilities due to misconfiguration. 3. If any known vulnerabilities are detected then this means non-compliance.
Test Type	Objective

Audit Item 13 – Business Security Policy

Reference	Own Contribution, Web Application Checklist (item ten)
Control Objective	To ensure that the business has a formal security policy defined and implanted.
Risk Area	Without a security policy there is no foundation or formal security objectives that have been signed off by executive management. The security function will have no power to implement security. WSS1. WA1.
Compliance	The web application security requirements must have been formally translated into a documented web application security policy.
Test Method	1. Ask to see evidence of a formally documented security policy that is aligned with business and security

	requirements.
	2. If the above does not exist or is inadequate in terms of scope and detail then this means non-compliance.
Test Type	Subjective

Audit Item 14 – Business Security Risk Assessment Process

Reference	Own Contribution, Web Application Checklist (item one)
Control Objective	To ensure that the business has addressed and are managing business security risks. The assessment should cover not only IT security but also business functionality security, business operations security, physical security, security policy framework and security management.
Risk Area	If the business has not identified business security risks and implemented a process to manage these risks, then the business is not in control of its security. WSS1. WA1.
Compliance	A formal web application security risk assessment must have been conducted and documented. There must be evidence that a process has been implemented to keep the risk assessment report up to date whenever there is a change to the business that impact the security of the application.
Test Method	1. Ask to see evidence of a formally documented business security risk assessment that included the web application.
	2. If the above does not exist or is inadequate in terms of scope and detail then this means non-compliance.
Test Type	Subjective

Audit Item 15 – Application Development Security Standards

Reference	Own Contribution
Control Objective	To ensure that the developers have a formal security standard to follow when developing and common security vulnerabilities are addressed at the development stage.

Risk Area	Without a formal web application security coding standard the likelihood that security vulnerabilities are not addressed will increase. This is especially important for inexperienced developers that may not know how to write secure applications. WA10.
Compliance	A web application security coding standard must be formally documented and actively used by the developers. A process must exist to ensure that the standard is kept up to date and covers the technologies and tools used by the business.
Test Method	1. Ask to see evidence of an adequate secure coding standard that is actively used by the application developers. 2. If the above does not exist or is inadequate in terms of scope and detail then this means non-compliance.
Test Type	Subjective

Audit Item 16 – Security Patching Process

Reference	Own Contribution
Control Objective	To ensure that there is a formal secure process for security patch release alerting, patch downloading, patch testing and patch implementation.
Risk Area	If patches are not applied to the operating systems and applications then attackers may exploit newly discovered vulnerabilities to compromise the security of the web application. WA12.
Compliance	A process for pro-actively researching new relevant vulnerabilities, securely obtaining patches, testing patches and implementing patches for the web application, the web server and its components must be formally documented and implemented.
Test Method	1. Ask to see evidence of how a recently relevant security patch was detected, securely obtained, tested and installed in the production environment according to a formally documented process. 2. If the above does not exist or is inadequate in terms of scope and detail then this means non-compliance.

Test Type	Subjective

Audit Item 17 – Information Gathering

Reference	Own Contribution, Web Application Checklist (item ten)
Control Objective	To ensure that the person responsible for web application security stays on top of new vulnerabilities and relevant news. This will ensure that any business security risk areas are properly addressed.
Risk Area	Unless the owner of the web application security is aware of new attack methodologies, new attack tools and new vulnerabilities then the business may be exposed to new potential problem areas and risks. WA4. REP1.
Compliance	A process must be implemented to allow the owner of the web application security to obtain relevant security information in order to be in the loop regarding any new developments within the web application security field.
Test Method	1. Ask to see evidence that the security owner has actively obtained any relevant recent information security information that is relevant to the business and the web application. 2. If the above does not exist or is inadequate in terms of scope and detail then this means non-compliance.
Test Type	Subjective

Audit Item 18 – Technical Architecture Diagram Update Process

Reference	Own Contribution, Web Application Checklist (item eleven)
Control Objective	To ensure that there is a process defined and implemented for keeping the technical architecture diagram up to date.
Risk Area	Without an up to date technical architecture diagram, it will be impossible to implement adequate security since the boundaries of the system, its components and their relationships have not been defined. WSS1. WA1. WA9.
Compliance	A process keeping the technical architecture diagram up to date must be formally documented and implemented. This process

	must be in line with the overall configuration management process for the business.
Test Method	1. Ask to see evidence of an up to date technical architecture diagram that is formally up dated in line with the business change management process. 2. If the above does not exist or is inadequate in terms of scope and detail then this means non-compliance.
Test Type	Subjective

Audit Item 19 – Security Ownership

Reference	Own Contribution
Control Objective	To ensure that formal ownership for the web application security has been established.
Risk Area	Without formal ownership of the web application security, there will be no central point of contact or driving force to make the web application secure.
Compliance	Security ownership of the web application must be formally assigned to an employee. This must be part of the formal job description. WSS1. WA1.
Test Method	1. Ask to see evidence that this responsibility has been formally assigned and documented and that this role is actively working to implement web application security. 2. If the above does not exist or is inadequate in terms of scope and detail then this means non-compliance.
Test Type	Subjective

Audit Item 20 – No Hidden Content

Reference	7.3. Auditing Web-based Applications (page 108), Web Application Checklist (item twenty-one)
Control Objective	To ensure that no hidden content that will aid attackers is published in the web application.
Risk Area	The first phase in any attack is the information gathering activity. Hidden content available in the web application may aid

	attackers in profiling the victim system.
Compliance	The business must have a formal process implemented for cleaning up all code before it is allowed to be published on the web server. WA4.
Test Method	1. Use a local web application proxy tool, log onto the application as a typical user. 2. Examine the client side source code for a random sample of web application pages. 3. Search for any potential hidden content that may be present. 4. If the above does not exist or is inadequate in terms of scope and detail then this means non-compliance.
Test Type	Objective

Audit Evidence

Completed Audit

Note! The auditor set up a local proxy, @stake Webproxy 1.0 (beta), on his laptop and pointed his web browser to this proxy to carry out the actual auditing. Once setup, Webproxy was configured to intercept all web requests to the audited website http://www.CompanyX.com. The default Webproxy configuration was applied, port 5111 for http traffic and 5112 for https traffic. Once Webproxy had been setup in this manner it was possible to examine and manipulate any request made to by the client-side browser to the web application server side. In other words, the auditor's requests could no longer be trusted by the web application server side since the auditor at any time could intercept and manipulate web requests.

For some web pages and web traffic @stake Webproxy 2.1 (evaluation) was used in this audit since the older version did not work reliably. The evaluation version does not support SSL and only allows the user to manipulate the first three fields. However, these limitations did not impact this audit. http://stake.com/products/webproxy/

Note! Screen shots have been converted to 256 color images to minimize the size of this document. All images have been anonymized.

Audit Item 1 - Validated Parameters

Control Objective	To ensure that information from web requests is validated before being used by a web application.
Compliance	The web application must block all input unless it is explicitly allowed. In other words, deny all input except known "good" input that is expected by the application. This must apply to any parameter that is passed to the web application.
Stimulus / Response	Yes
Actual Outcome	The auditor obtained a test account on the production system and logged on as a test user using a web browser. The auditor then proceeded to modify the personal profile and write a string of characters to the "title" field. The selected characters were characters that should not be accepted by the web application. The input characters should have been blocked by the web application according to security best practice. However, the web application happily accepted the following known problematic characters and stored them in the database: < > ; ! * /.. NULL

Assessment	The web application did not validate input and bad input was accepted. Forms were not protected by the web application.
Pass / Fail	Fail

Audit Item 2 - Secure Access Control

Control Objective	To ensure that restrictions on what authenticated users are allowed to do is properly enforced.
Compliance	The access control mechanism for the web application must be implemented according to the design and meet security requirements.
Stimulus / Response	Yes
Actual Outcome	The auditor obtained a test account on the production system and logged on as a test user. The auditor then examined what JSP pages were available in the support documentation, found on the manufacturer website, in the "Java Server Pages" section.

The auditor was able to successfully randomly access pages without following the described logical flow of pages, as described in the support documentation:

http://www.CompanyX.com/JSP/user.jsp
http://www.CompanyX.com/JSP/reg_b2b.jsp
http://www.CompanyX.com/JSP/reg_cons.jsp
http://www.CompanyX.com/JSP/CaseClose.jsp |
| Assessment | This implied that logged on users can freely access pages bypassing logical web page flow access control restrictions. Administrator functionality is incorrectly exposed to all users. |
| Pass / Fail | Fail |

Audit Item 3 - Secure Account Management

Control Objective	To ensure that account credentials are properly protected.
Compliance	The account management for the web application adequately protect the information and ensure that is not accessed by

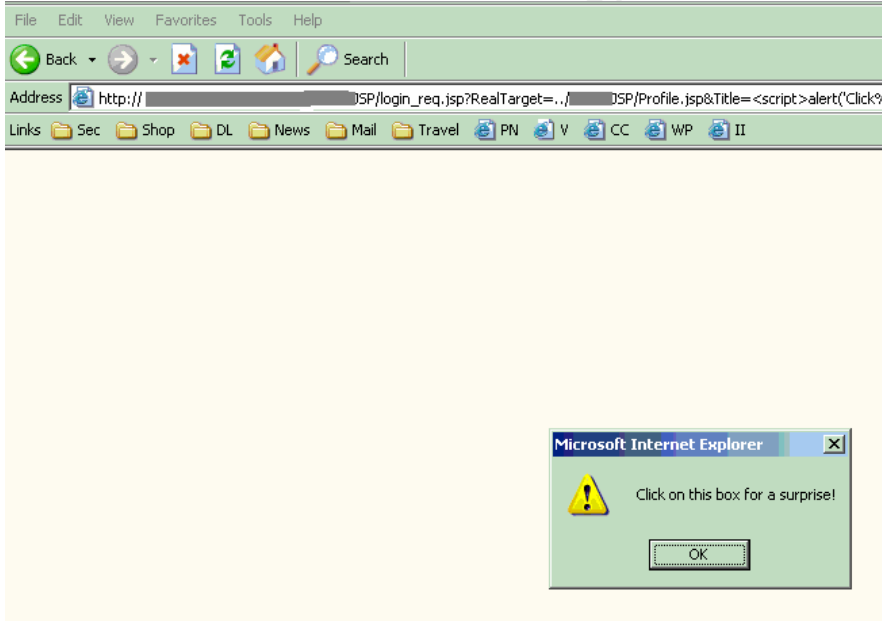|	
	unauthorised users.
Stimulus / Response	Yes
Actual Outcome	The auditor used @stake Webproxy 2.1 to manually examine the authentication mechanism, which turned out to be form-based. The variables and page names were obtained and used to configure the Brutus automated brute-forcing tool. Brutus was launched but the results were unreliable and inconsistent. The auditor proceeded to manually attempt ten consecutive failed logins for the same username followed by a successful login for the same username.
Assessment	The web application has no lock-out or slowdown mechanism to defend against repeated failed logon attempts or automated brute-force password cracking. This means non-compliance.
Pass / Fail	Fail

Audit Item 4 - Secure Session Management

Control Objective	To ensure that sessions are securely managed.
Compliance	Session management must be securely implemented to prevent sessions IDs from being predicted, brute forced or compromised in any way.
Stimulus / Response	Yes
Actual Outcome	The auditor used @stake Webproxy 2.1 to manually grab a collection of different session IDs from the web application server. The client side browser was shut down completely after each session ID had been grabbed. This was done in a rapid succession to get the following list of session IDs. 1. tM6hfJHeAQtgE79wl1TBorbC7Bu2QFtDp9588NPqJll7JFdcVq 2. tqgJ2v2tuFDm1uOc222xQmEszIeL62KjXC1N747khSvNj0fFtd 3. tU2DjkccqaUn3avtgXghJnpY2kgxQQbCGjiQLe4rHIL8VQQCK6 4. t2dH134Rvwl8j0SI6HZWCN4XsdB1Wuq1pkvD7rz2DNn21Cl3lJ 5. uXe1AFYCFsSuNcxgOwM1cTocV4gCWPYK3e57H6lg7fiKOYydgC 6. uBm6JfRvc5YxhYZBuwOiy11YLnf1whTf3NECdqQSIVjZGGZZ2J3

7. ugic02BAgKYiPChDDBCdO8J11sGt7f5cttGkzMKOU98133m1
    p1

8. uDFZCeI9ibFHqKWN5NoOhF6ib23CfdehDoPWzjZkdIMQQGA
    326

9. ulu5Z9VN2le3QM6xLRsbLCMJgIwhKo1SbrRycs8ryC5xC6N1
    GH

10. uDRvBy9Uu0yeM7L6bdmS1xAcaPDsykcwtmUaA2R2iqNdUxe
    lNj

11. ucGIJvOSBZZncVqI7q42FKVfSWjFcKoAfZUYoCVRuFpeoUU
    aXb

12. u399lSsgeq6Cu1xTTlDSY2KbnH8xxYtx7DVXXr152OTHH7Bxi
    i

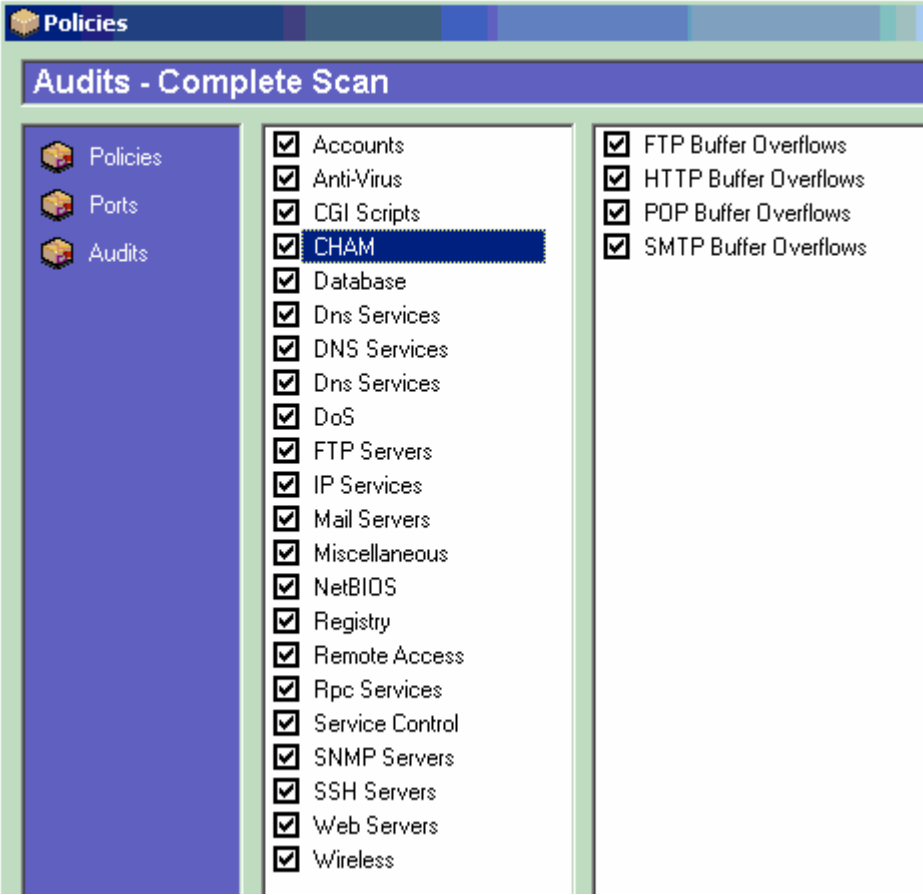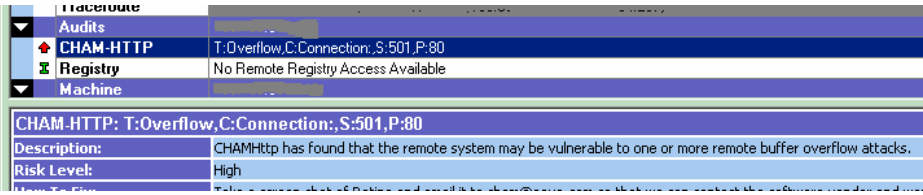|	
	The common prefix and suffix has been stripped away here for clarity, i.e. **JSESSIONID=15**u399lSsgeq6Cu1xTTlDSY2KbnH8xx Ytx7DVXXr152OTHH7Bxii!1690934490 **!XXXXXXXXXXXXXXXXXXXXXXXX** (the bold parts)
Assessment	The first alphanumeric in the session seems to be time dependent since the letter "t" slowly turns into "u" as time passes while sessions IDs are grabbed. However, simple analysis of the remaining session ID seems to indicate that this session ID scheme is secure and can not be predicted.
Pass / Fail	Pass

Audit Item 5 - Cross-Site Scripting (XSS) Controls

Control Objective	Ensure that the web application can not be used as a mechanism to transport an attack to an end user's browser.
Compliance	The web application must not allow code to be injected and displayed back in the users' browsers.
Stimulus / Response	Yes
Actual Outcome	The auditor obtained a test account on the production system and logged on as a test user using @stake Webproxy 2.1 to intercept all web all traffic to the audited website. The auditor navigated to numerous pages in the web application, which looked like potential XSS targets and injected the following java script code into various web application variables:

	`<script>alert('Click%20on%20this%20box%20for%20a %20surprise!')</script>`
	This did not result in successful cross site scripting compromises. However, at the end of this test the auditor injected the same code into the "Title" variable on the following page:
	http://www.CompanyX.com/JSP/login_req.jsp? RealTarget=../JSP/Profile.jsp&Title= Customer%20Profile
	The result is shown in the screen shot below.
Assessment	Only one cross scripting vulnerability was found. However this means non-compliance.
Fail / Pass	Fail

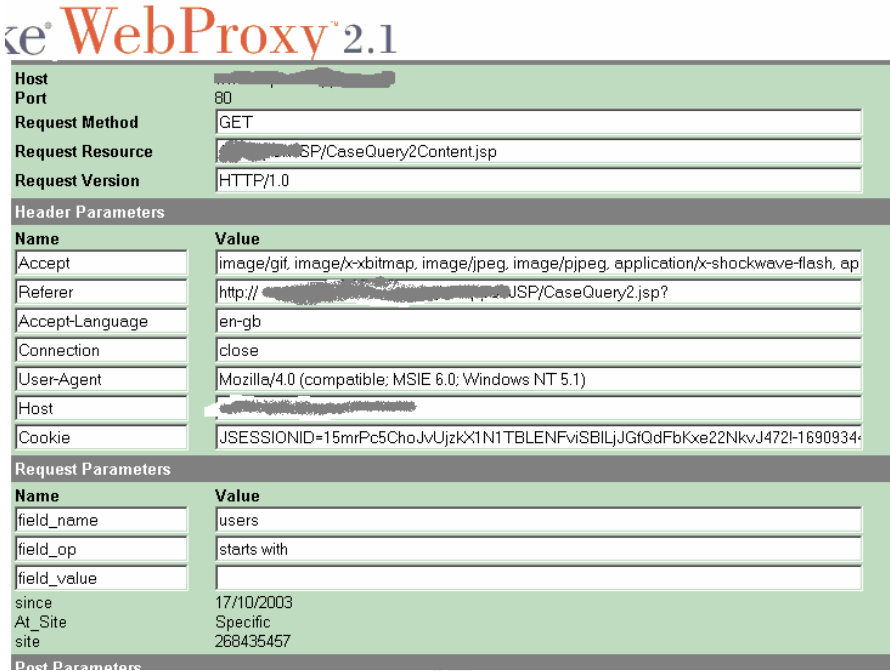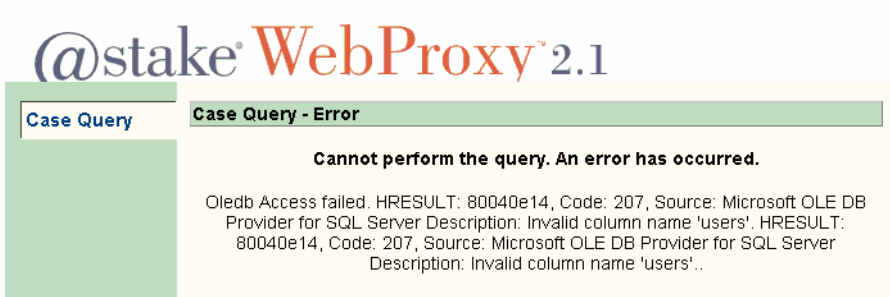## Audit Item 6 - Buffer Overflow Controls

Control Objective	Ensure that the web application components are not vulnerable to buffer overflow attacks.
Compliance	None of the web application components must be vulnerable to buffer overflow attacks.
Stimulus /	Yes

Response	
Actual Outcome	The auditor did not have access to the web application source code and a code buffer overflow review was not possible. However, the Retina vulnerability scanner was launched to do a complete scan including buffer overflow probing using the eEye Common Hacking Attack Modules (CHAM). See screenshot below for Retina scan settings. **Policies** **Audits - Complete Scan** Policies Ports Audits ☑ Accounts ☑ FTP Buffer Overflows ☑ Anti-Virus ☑ HTTP Buffer Overflows ☑ CGI Scripts ☑ POP Buffer Overflows ☑ CHAM ☑ SMTP Buffer Overflows ☑ Database ☑ Dns Services ☑ DNS Services ☑ Dns Services ☑ DoS ☑ FTP Servers ☑ IP Services ☑ Mail Servers ☑ Miscellaneous ☑ NetBIOS ☑ Registry ☑ Remote Access ☑ Rpc Services ☑ Service Control ☑ SNMP Servers ☑ SSH Servers ☑ Web Servers ☑ Wireless
Assessment	The scanning result returned a potential buffer overflow vulnerability in the web application on port 80. This needs to be investigated, see screenshot below.. Traceroute **Audits** CHAM-HTTP T:Overflow,C:Connection:,S:501,P:80 Registry No Remote Registry Access Available **Machine** CHAM-HTTP: T:Overflow,C:Connection:,S:501,P:80 Description: CHAMHttp has found that the remote system may be vulnerable to one or more remote buffer overflow attacks. Risk Level: High How To Fix: Take a screen shot of Retina and email it to cham@eeye.com so that we can contact the software vendor and ... This means non-compliance until the web application team has

	proven that this detected buffer overflow is a false positive.
Fail / Pass	Fail

Audit Item 7 - Command Injection Controls

Control Objective	To ensure that the web application is not vulnerable to command injection attacks.
Compliance	None of the web application components must be vulnerable to code injection attacks.
Stimulus / Response	Yes
Actual Outcome	The auditor obtained a test account on the production system and logged on as a test user using @stake Webproxy 2.1 to intercept all web all traffic to the audited website. The auditor changed the following variables and posted them to the page below: Name: "field_name" Value: "id_number" -> changed to -> "users" http://www.CompanyX.com/JSP/CaseQuery2Content.jsp
Assessment	The outcome was a verbose error message indicating that this page is vulnerable to SQL injection attacks. A patient attacker can start to map the data base by manipulating the variables that are passed to the JSP page in question and analyzing the returned error messages. Screen shot of modified variables (variable field_name changed to users):

Screen shot of resulting SQL server verbose error message:
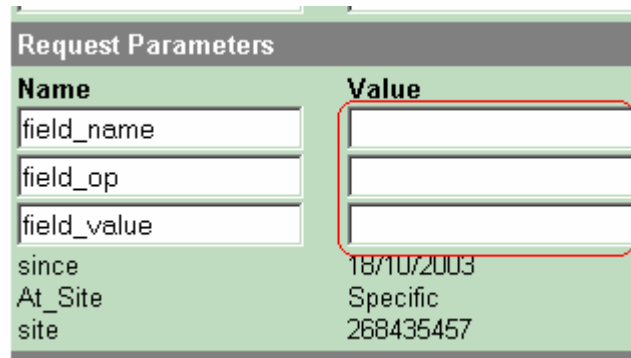

Fail / Pass	Fail

Audit Item 8 – Secure Error Handling

Control Objective	To ensure that web application error conditions that occur during normal operation are handled properly.
Compliance	The web application error messages must not leak any information that can aid attackers in compromising the security of web application.
Stimulus / Response	Yes
Actual Outcome	The auditor obtained a test account on the production system and logged on as a test user using @stake Webproxy 2.1 to intercept

all web all traffic to the audited website.

The auditor navigated to the query page and simply removed all the values from the variables and posted empty variables to the page:

Screen shot of empty variables that were modified on the fly in Webproxy.

Page:
http://www.CompanyX.com/JSP/CaseQuery2.jsp

Screen shot of the resulting server error response, "Syntax Error in string Desc< ??? >".
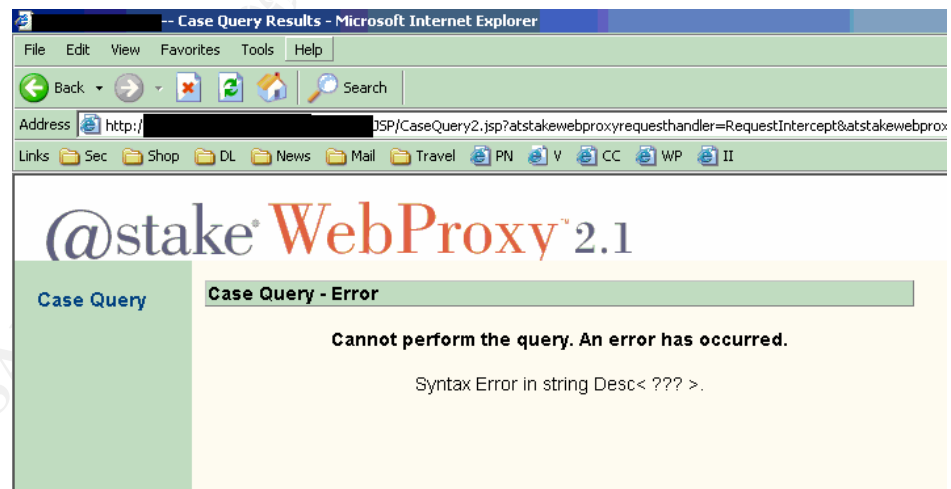
Changing the variable values resulted in different error messages.

Assessment	It was possible to force the web application into an error state, which revealed information that may aid attackers who are interested in profiling the system for further malicious activity and abuse.

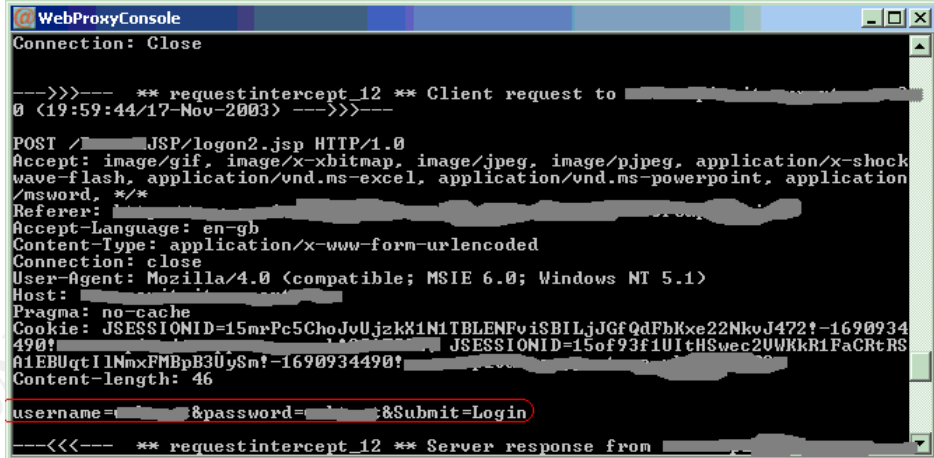|	
Fail / Pass	Fail

Audit Item 9 - Secure Use of Cryptography

Control Objective	To ensure that the web applications use cryptographic functions to protect information and credentials, where required. **Note!** This audit item is limited to the use of cryptographic controls within the web application. Cryptographic controls for data stored in the back-end data base is out of scope).
Compliance	The web application must have implemented cryptographic controls securely according to security requirements.
Stimulus / Response	Yes
Actual Outcome	The auditor obtained a test account on the production system and logged on as a test user using @stake Webproxy 2.1 to intercept all web all traffic to the audited website. The login process was examined for use of cryptographic secure sockets layer (SSL) support. Screenshot of the login page http request. Variables posted in clear-text: Username = <username> Password = <password> Submit = Login The auditor proceeded to step through all remaining pages in the web application but no SSL enabled pages were found.

Assessment	SSL was not implemented to protect the logon credentials. The credentials were passed with a POST statement using three variables sent in clear text.
Fail / Pass	Fail

Audit Item 10 – Secure Remote Administration

Control Objective	To ensure that the web application allows administrators to securely remotely administer the web application.
Compliance	All web application remote administration interfaces must be locked down according to security requirements.
Stimulus / Response	Yes
Actual Outcome	The auditor did not have access to the required remote administration client software or tools for auditing the security of the remote admin interfaces.
Assessment	Not possible.
Pass / Fail	N/A

Audit Item 11 – Secure Web Application Configuration

Control Objective	To ensure that the web application configuration security standard security meets the defined security requirements and follows security best practice.
Compliance	The web application must be configured securely to meet security requirements and the configuration must be under strict configuration management control.
Stimulus / Response	Yes
Actual Outcome	The web application server was scanned with the eEye Retina and Foundstone Foundscan web application vulnerability audits. The tools were updated with all the latest vulnerability signatures at the time the scans were performed. Please see Appendix A & B for the results of the scan.
Assessment	Foundscan did not find any web application vulnerabilities.

	However, Retina identified a potential web application buffer overflow vulnerability (see audit item 6). Until this potential buffer overflow vulnerability has been proven to be a false positive – this means non-compliance.
Pass / Fail	Fail

Audit Item 12 – Secure Web Server Configuration

Control Objective	To ensure that the web server configuration security standard security strong.
Compliance	The web server must be configured securely to meet security requirements and the configuration must be under strict configuration management control.
Stimulus / Response	Yes
Actual Outcome	The web application server was fully scanned on all TCP and UDP ports 1-65335 using the eEye Retina and the Foundstone Foundscan vulnerability scanners. The tools were updated with all the latest vulnerability signatures at the time the scans were performed. The scans included all vulnerability scans with the exception of aggressive policy or intrusive scan options. Please see Appendix A & B for the results of the scan.
Assessment	None of the tools identified any high risk vulnerabilities. However, the following ports should not have been open on the external firewall in front of the web application – only port 80 (http) and 443 (https) are required to be open. Foundscan reported open ports: netbios-ssn - NETBIOS Session ServicePort: tcp - 139 ms-sql-s - Microsoft-SQL-ServerPort: tcp - 1433 ica - Citrix ICAPort: tcp - 1494 ncube-lm - nCube License ManagerPort: tcp - 1521 pdap-np - Prospero Data Access Prot non-privPort: tcp - 1526 h323hostcall - h323hostcallPort: tcp - 1720 - bmc-patrol-agentPort: tcp - 3300 ms-termsrv - Microsoft Terminal ServerPort: tcp - 3389 pcanywheredata - pcANYWHEREdataPort: tcp - 5631 javaWS - Sun JavaWebServer over SSLPort: tcp - 7070

	http-alt - HTTP Alternate (see port 80)Port: tcp - 8080
	Retina reported open ports: 139: NETBIOS-SSN - NETBIOS Session Service 1416: NOVELL-LU6.2 - Novell LU6.2 1418: TIMBUKTU-SRV2 - Timbuktu Service 2 Port 1433: MS-SQL-S - Microsoft-SQL-Server 1521: NCUBE-LM - nCube License Manager 1526: PDAP-NP - Prospero Data Access Prot non-priv 1720: No name 1801: No name 2059: No name 2101: No name 2103: ZEPHYR-CLT - Zephyr Serv-HM Conncetion 2105: EKLOGIN - Kerberos (v4) Encrypted RLogin 2157: No name 3181: No name 3200: No name 3300: No name 3389: MS RDP (Remote Desktop Protocol) / Terminal Services 3399: No name 7070: ARCP (nasa.gov) 8080: Generic - Shared service port / HTTP Alternate 8091: SIMPLE, SECURE WEB SERVER 1.1 9100: JETDIRECT - HP JetDirect Card 9999: DISTINCT - distinct 20200: No name The web application server is not protected adequately by the firewall and this means non-compliance.
Pass / Fail	Fail

Audit Item 13 – Business Security Policy

Control Objective	To ensure that the business has a formal security policy defined and implanted.
Compliance	The web application security requirements must have been formally translated into a documented web application security policy.
Stimulus / Response	No
Actual	The auditor asked to be provided with evidence of a formally

Outcome	documented web application security policy, aligned with business and security requirements.
Assessment	A web application security policy was not available.
Pass / Fail	Fail

Audit Item 14 – Business Security Risk Assessment Process

Control Objective	To ensure that the business has addressed and are managing business security risks. The assessment should cover not only IT security but also business functionality security, business operations security, physical security, security policy framework and security management.
Compliance	A formal web application security risk assessment must have been conducted and documented. There must be evidence that a process has been implemented to keep the risk assessment report up to date whenever there is a change to the business that impact the security of the application.
Stimulus / Response	No
Actual Outcome	The auditor asked to be provided with evidence of a formally documented business security risk assessment that addressed web application security.
Assessment	A business security risk assessment of web application security was not available and there was no process implemented to keep the risk assessment up to date.
Pass / Fail	Fail

Audit Item 15 – Application Development Security Standards

Control Objective	To ensure that the developers have a formal security standard to follow when developing and common security vulnerabilities are addressed at the development stage.
Compliance	A web application security coding standard must be formally documented and actively used by the developers. A process must exist to ensure that the standard is kept up to date and

	covers the technologies and tools used by the business.
Stimulus / Response	No
Actual Outcome	The auditor asked to be provided with evidence of a formally documented web application security coding standard that incorporates security best practice.
Assessment	A web application security coding standard that incorporates security best practice was not available.
Pass / Fail	Fail

Audit Item 16 – Security Patching Process

Control Objective	To ensure that there is a formal secure process for security patch release alerting, patch downloading, patch testing and patch implementation.
Compliance	A process for pro-actively researching new relevant vulnerabilities, securely obtaining patches, testing patches and implementing patches for the web application, the web server and its components must be formally documented and implemented.
Stimulus / Response	No
Actual Outcome	The auditor asked to be provided with evidence of how a recently relevant security patch was detected, securely obtained, tested and installed in the production environment according to a formally documented process.
Assessment	There was no formal patching process implemented.
Pass / Fail	Fail

Audit Item 17 – Information Gathering

Control Objective	To ensure that the person responsible for web application security stays on top of new vulnerabilities and relevant news. This will ensure that any business security risk areas are properly addressed.
Compliance	A process must be implemented to allow the owner of the web

	application security to obtain relevant security information in order to be in the loop regarding any new developments within the web application security field.
Stimulus / Response	No
Actual Outcome	The auditor asked to be provided with evidence that the web application security owner has actively obtained any relevant recent security information, relevant to the business and the web application and put this information to use in form of improvements to web application security.
Assessment	There was no evidence available that the owner of web application security proactively stays up to date with relevant security information and applies this knowledge in day to day security work.
Pass / Fail	Fail

Audit Item 18 – Technical Architecture Diagram Update Process

Control Objective	To ensure that there is a process defined and implemented for keeping the technical architecture diagram up to date.
Compliance	A process keeping the technical architecture diagram up to date must be formally documented and implemented. This process must be in line with the overall configuration management process for the business.
Stimulus / Response	No
Actual Outcome	The auditor asked to be provided with evidence of an up to date technical architecture diagram that is formally up dated in line with the business change management process.
Assessment	An up to date diagram was produced in an ad-hoc fashion in a couple of days' time after the auditor made his request. This was not a formal process.
Pass / Fail	Fail

Audit Item 19 – Security Ownership

Control	To ensure that formal ownership for the web application security

Objective	has been established.
Compliance	Security ownership of the web application must be formally assigned to an employee. This must be part of the formal job description.
Stimulus / Response	Yes
Actual Outcome	The auditor asked to be provided with evidence that web application security responsibility had been formally assigned to a role, documented and that this job role is actively working to implement web application security.
Assessment	There was no evidence that web application security responsibilities had been formally incorporated in a job description and ownership had been clearly assigned.
Pass / Fail	Fail

Audit Item 20 – No Hidden Content

Control Objective	To ensure that no hidden content that will aid attackers is published in the web application.
Compliance	The business must have a formal process implemented for cleaning up all code before it is allowed to be published on the web server.
Stimulus / Response	Yes

Actual Outcome	The auditor navigated through the site with the @stake Webproxy v2.1 tool and recorded the html source code output. The source code was examined for any hidden content or information leakage.
	
	1. Comments are present on the homepage http://www.CompanyX.com:
	
	// Copyright XXXXXXXXXX. All rights reserved. ProductX is a
	// registered trademark, and ProductX XXXXXX, ProductX XXXXX and ProductX XXXXXXX are
	// trademarks of XXXXXXXXXX. All other products or services mentioned herein are the
	// property of their respective owners and should be treated as such.

// Description: required field list object with type 'And', in which every field is required.
// Return: none

2. Validation code is present on the client-side on the homepage http://www.CompanyX.com:

```
// Description: Checks the length of each non-required field
// Return: true if a field exceeds its maximum length/false if not
function NonRequiredFieldListCheckFieldLength()
{
        for (i = 0; i < this.fieldList.length; i++)
    {
                if(this.fieldList[i].CheckFieldLength() == false)
                {
                    var strFmt = "";
                    if (numExceeded > 1)
                            strFmt = "The value in %1 exceeds the
maximum length by %2 bytes.";
                    else
                            strFmt = "The value in %1 exceeds the
maximum length by %2 byte.";
                    alert(strMsg);
                    return false;
                }
        }

        return true;
}
```

3. More validation code resent on the client-side on the page http://www.CompanyX.com/JSP/PortalCompanyX.jsp:

```
// Description: used to perform the validation for a field
// Return: true/false
function RequiredFieldValidate()
{
        // left trim
        var numPos = 0;
        var strValue = new String(this.field.value);

        if(strValue.length == 0)
                return false;

        while ((strValue.charAt(numPos) == " ") ||
```

```
(strValue.charAt(numPos) == "\t")) {
            numPos++;
    }
    if (numPos > 0) {
            strValue = strValue.substr(numPos);
    }

    // right trim
    numPos = strValue.length - 1;

    while ((strValue.charAt(numPos) == " ") ||
(strValue.charAt(numPos) == "\t")) {
            numPos--;
    }
    if (numPos < strValue.length - 1) {
            strValue = strValue.substring(0, numPos + 1);
    }

    if(strValue == "")
            return false;
    else
            return true;
}
```

4. Another example of validation code on the client-side on the page
http://www.CompanyX.com/JSP/PortalCompanyX.jsp:

```
// Description: Checks if a required field exceeds its maximum
length
// Return: True if length does not need to be checked or if it does
not exceed length/false if

exceeds
// its maximum length
function RequiredFieldCheckFieldLength()
{
   if (this.length == -1)
     return true;

     // left trim
     var numPos = 0;
     var strValue = new String(this.field.value);

     if(strValue.length == 0)
             return true;
```

```
            while ((strValue.charAt(numPos) == " ") ||
(strValue.charAt(numPos) == "\t")) {
                numPos++;
            }
            if (numPos > 0) {
                strValue = strValue.substr(numPos);
            }

            // right trim
            numPos = strValue.length - 1;

            while ((strValue.charAt(numPos) == " ") ||
(strValue.charAt(numPos) == "\t")) {
                numPos--;
            }
            if (numPos < strValue.length - 1) {
                strValue = strValue.substring(0, numPos + 1);
            }
    var numBytes   = 0;
    var TestString = new String(strValue);
    var i = 0;
    for (; i < TestString.length; i++)
    {
       var numCharVal = TestString.charCodeAt(i);
       if (numCharVal < NumMaxBytes)
         numBytes   = numBytes + 1;
       else
         numBytes   = numBytes + 2;
    }
    var numFieldLength = this.length;
    if (numBytes > numFieldLength)
    {
        numExceeded = numBytes - numFieldLength;
       return false;
    }
    numExceeded = 0;
    return true;
}
            // right trim
            numPos = strValue.length - 1;

            while ((strValue.charAt(numPos) == " ") ||
(strValue.charAt(numPos) == "\t")) {
                numPos--;
            }
```

| | ```
if (numPos < strValue.length - 1) {
        strValue = strValue.substring(0, numPos + 1);
}

if(strValue == "")
        return false;
else
        return true;
}
``` |
| **Assessment** | There are comments splattered all over the html source code. This may aid attackers when they profile the web application and plan their attacks.

In addition there is client side java script code on virtually all web pages. All client side requests can be intercepted and manipulated by any user who is running a local proxy tool. This means that any validation that takes place on the client-side can be by-passed by modifying variables to any desired valued before being submitting them back to web application server side. This means non-compliance. |
| **Pass / Fail** | Fail |

Residual Risk Measure

Given that only one control objective was achieved while eighteen were not and one could not be properly audited, the overall residual risk of the audited web application is high for CompanyX.

The majority of the vulnerabilities can be mitigated by applying a security best practice methodology to define security requirements, identify security risk areas, select appropriate counter measures, document and implement these countermeasures. A security risk assessment process should also be defined and implemented to ensure that any new risks to the web application are addressed when new vulnerabilities appear or fundamental changes that impact the security of the application are made. All the above can be considered accepted industry practice for the type and size of business that we are dealing with for this audit. Not doing the above places CompanyX in a situation where they are not in control of business security risks and also expose their own and their customer's information to be abused by attackers. Not to mention the reputational damage and potential loss of customer confidence.

In addition, the web application developers can mitigate most of the common web application vulnerabilities by following best practice as outlined in the "OWASP Guide to Building Secure Web Applications".

The exception here is the potential buffer overflow vulnerability discovered in the audit. This could potentially be in the vendor binaries, which can not be mitigated by CompanyX. The only way this can be mitigated is for the vendor to release a patch that addresses the buffer overflow.

Estimated work effort to reduce the residual risk to an acceptable level breaks down as follows.

Work Activity	Estimated work effort (man days)	Required Resource
Risk Assessment Of Web Application and definition of security requirements.	2	External Security Consultant
	0.5	Internal Business Manager
Web Application Security Workshop	4	Internal Web Application Developers
	2	External Security Consultant
Security Policies and Procedures Drafting	2	External Security Consultant
	1	Internal Web Application Developers
	1	Internal System Administrator
	1	Internal Business Manager
Web Application Code Review and Update	1	External Security Consultant
	10	Internal Web Application Developers
Re-Audit	1	External Security Consultant
Total	**25.5**	

It is up to CompanyX to make a business decision, whether to do nothing and accept the residual risk level highlighted by the results of the audit or to go ahead and spend the required man days to mitigate the risks to levels the business finds acceptable.

System auditing properties

All the steps taken to this point clearly demonstrate that the CompanyX web application can be audited using a combination of objective and subjective compliance tests. The technical controls can be objectively audited with various stimulus and response techniques and tools while policies and procedures can be examined and judged subjectively by a subject matter expert.

There will always be new attack methodologies and exploits of freshly discovered vulnerabilities and that why it is important for CompanyX to start considering security as a never-ending process and ensure that new risks are assessed and mitigated by selecting and implementing appropriate countermeasures if required.

One of the checklist items that could not be properly audited was the remote management administration security but this can be done in the future if the auditor is provided with the required information and tools by CompanyX. The only item that could not be successfully audited was the buffer overflow vulnerability, which is likely to be a proprietary vendor binary executable that can not be inspected by the auditor or CompanyX.

Audit Report

Executive summary

Overall the audit objectives were achieved. The CompanyX web application and web application server were audited against accepted industry best practice with regards to business security with a focus on the technical aspects. Twenty checklist items were audited and only one received a passing mark. Overall the residual risk level is high for the audited web application and web application server. The audit result gives clear evidence that security has not been properly addressed with regards to the planning, design, implementation and maintenance of the CompanyX web application and web server.

Audit findings

Below is a summary of the audit findings. For more detail please refer back to the Audit section in this paper and cross reference using the numbers in the first table column. Out of twenty audited checklist items, eighteen fails, one pass and one "not applicable" were noted. This gives an overall pass percentage of roughly five percent, which indicates a high residual risk level. The results from this audit clearly give evidence indicating that appropriate security methodology and industry accepted web application security best practice has not been followed or applied at CompanyX.

Ref	Audit Item	Test	Outcome (Pass/Fail)
1	Validated Parameters	@stake Webproxy web request and parameter manipulation	Fail
2	Secure Access Control	Brutus / Manual password auditing	Fail
3	Secure Account Management	@stake Webproxy web request and parameter manipulation	Fail
4	Secure Session Management	@stake Webproxy intercept and manual analysis of session IDs	Pass
5	Cross-Site Scripting (XSS) Controls	@stake Webproxy web request and manual code injection	Fail
6	Buffer Overflow Controls	eEye Retina automated buffer	Fail

Ref	Audit Item	Test	Outcome (Pass/Fail)
		overflow vulnerability scan	
7	Command Injection Controls	@stake Webproxy web request and manual code injection	Fail
8	Secure Error Handling	@stake Webproxy web request and manual code injection	Fail
9	Secure Use of Cryptography	@stake Webproxy	Fail
10	Secure Remote Administration	Remote Admin Tool attack	N/A
11	Secure Web Application Configuration	eEye Retina & Foundstone Foundscan automated vulnerability scan	Fail
12	Secure Web Server Configuration	eEye Retina & Foundstone Foundscan automated vulnerability scan	Fail
13	Business Security Policy	Interview and request for documentation	Fail
14	Business Security Risk Assessment Process	Interview and request for documentation	Fail
15	Application Development Security Standards	Interview and request for documentation	Fail
16	Security Patching Process	Interview and request for documentation / eEye Retina automated vulnerability scan	Fail
17	Information Gathering	Interview and request for documentation	Fail
18	Technical Architecture Diagram Update Process	Interview and request for documentation	Fail
19	Security Ownership	Interview and request for documentation	Fail

Ref	Audit Item	Test	Outcome (Pass/Fail)
20	No Hidden Content	@stake Webproxy source code output file recording	Fail

Background/risk

CompanyX has not defined security requirements or performed a security risk assessment on their web application portal that they offer as a service to their customers. This means that CompanyX is not in control of business security risk. The assets to be protected have not been formally identified and no formal business decision has been made with regards to the appropriate selection, implementation and maintenance of required security controls.

This is not smart business and may also give a false sense of security - if time and effort is spent on putting security controls in place that are in effect useless, with regards to real risks that need to be mitigated. For example, the most expensive and perfectly configured firewall would do nothing to mitigate against many of the vulnerabilities found in this audit, since these are found in the web application. The firewall must be opened up to let the web application traffic flow and enable communication with customer web browsers.

Many of the web application vulnerabilities may seem low risk, however when combined in a clever way and used in combination, these flaws may become high risk and can be exploited to compromise the entire web application and even the business assets that are located in the tier behind the web application. For example, an attacker can exploit the verbose error messages to map the structure of the data base tables and manipulate the web application to return or even modify data stored in the data base.

A successful attack will not only incur cost in terms of internal resources required to containment and response but is also likely to negatively affect customer confidence and even the CompanyX share price if an incident made the headline news in the television or print media.

Audit recommendations

In order to address the root of the problems and residual risk identified in this audit a business security risk management process should be put in place. This will ensure that relevant web application security risks are identified and mitigated to a level, which is acceptable by CompanyX. Once the security requirements have been established these should be translated into a web application security policy which needs to be signed-off and fully supported by senior management. CompanyX also needs to assign security ownership in the area of web application security.

In addition, the web application developers need to spend time and become familiar with common web application vulnerabilities and what can be done to address these weaknesses. If needed, a web application security subject matter expert should be brought in to ensure that the required knowledge is transferred to the developers. The focus should first be on the principles that need to be applied in order to develop secure applications and once this has been accomplished the developers should take these principles and drill down into details relevant to the specific technology CompanyX use in its implementation. All this work should be formally captured in a CompanyX secure coding policy and web application security standards.

The security procedures around maintenance and support of the web application and the web server should also be reviewed and formally documented in line with the security policies. Important areas include patching and a process that ensures that the technical documentation such as architecture diagrams is kept up to date in line with the CompanyX change control management process.

Cost Consideration

The estimated cost to implement the recommendations outlined in the Audit Recommendation section above has been estimated at a total of 25.5 man days. Seven of these need to be done by a subject matter expert, such as an external security consultant, while the remaining eighteen and a half days will be internal resources. For more information please refer to the Residual Measure section in the previous chapter.

Compensating controls

The likelihood that the CompanyX web application will be compromised is high since most common web application vulnerabilities are present. The auditor believes it is not a matter of if but when the application will be subverted by an attacker. The vulnerabilities identified in this audit can be exploited with a regular web browser running a web proxy to intercept and manipulate the web requests. Specialised tools are freely available to download on the internet. White papers and presentations that explain in detail how to exploit the vulnerabilities identified in this audit can also easily be found using a search engine such as Google.

It is now up to CompanyX to make a business decision regarding what to do next. One option is to simply accept the current residual risk and do nothing. The recommended option is to consider the risks and vulnerabilities discussed in this paper and the cost of implementing the recommendations that have come out of this audit.

This auditor strongly suggests that CompanyX review their security requirements, implements a simple risk management process, create a security policy and

supporting security management procedures. CompanyX also need to allocate time for their developers to read and work with the OWASP Guide to Building Secure Web Application and apply this newfound knowledge to remove the present vulnerabilities from the CompanyX web application and create a formal secure coding standard. It would also be advisable to have the developers test drive their own application with a tool such as @stake Webproxy and show how easily web requests can be maliciously manipulated.

Appendix A – Foundstone Foundscan result

```
FoundScan for CompanyX: Network Services Report
Report Generated: 11-14-2003 15:32:01 GMT Standard Time

Navigate
Summary
FoundScore
Network Map
Discovered Hosts
Operating Systems
Network Services
Vulnerability Trend - Short Term
Web Module
Banners
Configuration
History


CompanyX
Scan Name: www.CompanyX.com

The graph and table below identify all services discovered during the
scan. For information regarding the type of service discovered and its
inherent risks, you may click on the name of the service in the far
left column for further details.

http - World Wide Web
HTTPPort: tcp - 80
IP AddressMachine NameBannerConnect
XXX.XXX.XXX.XXX[unavailable][view][open]

netbios-ssn - NETBIOS Session ServicePort: tcp - 139
IP AddressMachine NameBannerConnect
XXX.XXX.XXX.XXX[unavailable]n/an/a

ms-sql-s - Microsoft-SQL-ServerPort: tcp - 1433
IP AddressMachine NameBannerConnect
XXX.XXX.XXX.XXX[unavailable]n/an/a

ica - Citrix ICAPort: tcp - 1494
IP AddressMachine NameBannerConnect
XXX.XXX.XXX.XXX[unavailable]n/an/a

ncube-lm - nCube License ManagerPort: tcp - 1521
IP AddressMachine NameBannerConnect
XXX.XXX.XXX.XXX[unavailable]n/an/a

pdap-np - Prospero Data Access Prot non-privPort: tcp - 1526
IP AddressMachine NameBannerConnect
XXX.XXX.XXX.XXX[unavailable]n/an/a


h323hostcall - h323hostcallPort: tcp - 1720
IP AddressMachine NameBannerConnect
```

XXX.XXX.XXX.XXX[unavailable]n/an/a

- bmc-patrol-agentPort: tcp - 3300
IP AddressMachine NameBannerConnect
XXX.XXX.XXX.XXX[unavailable]n/an/a

ms-termsrv - Microsoft Terminal ServerPort: tcp - 3389
IP AddressMachine NameBannerConnect
XXX.XXX.XXX.XXX[unavailable]n/an/a

pcanywheredata - pcANYWHEREdataPort: tcp - 5631
IP AddressMachine NameBannerConnect
XXX.XXX.XXX.XXX[unavailable]n/an/a
javaWS - Sun JavaWebServer over SSLPort: tcp - 7070
IP AddressMachine NameBannerConnect
XXX.XXX.XXX.XXX[unavailable][view]n/a

http-alt - HTTP Alternate (see port 80)Port: tcp - 8080
IP AddressMachine NameBannerConnect
XXX.XXX.XXX.XXX[unavailable][view]n/a

(c)2000-2001 Foundstone Inc.

FoundScan for CompanyX: Web Application Assessment Report
Report Generated: 11-14-2003 15:32:10 GMT Standard

Navigate
Summary
FoundScore
Network Map
Discovered Hosts
Operating Systems
Network Services
Vulnerability Trend - Short Term
Web Module
Banners
Configuration
History

CompanyX
Scan Name: ProductX - www.CompanyX.com

Summary of Servers Scanned (1 total)

Web ServersTypeProtocolPort
[Unknown]XXX.XXX.XXX.XXX WebLogic
WebLogic Temporary Patch for CR064988 02/05/2002 15:38:23http80

Web authentication testing was enabled. No accounts were found to be
vulnerable during this scan.

Smart GuessWork was enabled. No vulnerabilities were discovered by the
Smart GuessWork feature during this scan.

Source Code Disclosure was enabled. No vulnerabilities were discovered
by the Source Code Disclosure feature during this scan.

SQL Security Analysis was enabled. No vulnerabilities were discovered
by the SQL Security Analysis feature during this scan.

(c)2000-2001 Foundstone Inc.

FoundScan for CompanyX: Banners Report
Report Generated: 11-14-2003 15:32:03 GMT Standard Time

Navigate
Summary
FoundScore
Network Map
Discovered Hosts
Operating Systems
Network Services
Vulnerability Trend - Short Term
Web Module
Banners
Configuration
History

CompanyX
Scan Name: www.CompanyX.com

IP AddressPortBanner
XXX.XXX.XXX.XXX
80--> GET / HTTP/1.0
Host: XXX.XXX.XXX.XXX

HTTP/1.0 302 Moved Temporarily
Date: Fri, 14 Nov 2003 15:21:54 GMT
Location: http://XXX.XXX.XXX.XXX/index.html
Server: WebLogic WebLogic Temporary Patch for CR064988 02/05/2002
15:38:23

Content-Type: text/plain
Connection: Close
XXX.XXX.XXX.XXX
8080--> GET / HTTP/1.0
Host: XXX.XXX.XXX.XXX

[Connection closed by remote host]
XXX.XXX.XXX.XXX
7070--> GET / HTTP/1.0
Host: XXX.XXX.XXX.XXX

[Connection closed by remote host]

(c)2000-2001 Foundstone Inc.

Appendix B – eEye Retina result

http://www.eeye.com/html/Products/Retina/index.html

General: XXX.XXX.XXX.XXX

Address: XXX.XXX.XXX.XXX
No More Details Available

Report Date: 11/18/03 03:34:51 AM
No More Details Available

Domain Name: unknown
No More Details Available

Ping Response: Host Responded
No More Details Available

Avg Ping Response: 1025 ms
No More Details Available

Time To Live: 120
No More Details Available

Traceroute: 10.0.0.1, XXX.XXX.XXX.XXX, XXX.XXX.XXX.XXX, XXX.XXX.XXX.XXX, XXX.XXX.XXX.XXX, XXX.XXX.XXX.XXX, XXX.XXX.XXX.XXX, XXX.XXX.XXX.XXX, XXX.XXX.XXX.XXX,XXX.XXX.XXX.XXX
No More Details Available

Audits: XXX.XXX.XXX.XXX

CHAM-HTTP: T:Overflow,C:Connection:,S:501,P:80

Risk Level: High

Description: CHAMHttp has found that the remote system may be vulnerable to one or more remote buffer overflow attacks.

How To Fix:

Take a screen shot of Retina and email it to cham@eeye.com so that we can contact the software vendor and work with them to create a fix. If possible, select the Create Log option under Retina's Tools->options menu. Rerun retina against this host (after starting the HTTP server on the remote machine). This creates a log in your top retina directory call RETDEBUG.LOG. This log will better help us diagnose the problem in the HTTP server.

CVE: GENERIC-MAP-NOMATCH

Registry: No Remote Registry Access Available

Risk Level: Information

Description: This alert is only to notify you that Retina was not able to access the remote system's registry. Without registry access, Retina will still be able to remotely audit for vulnerabilities, although having access to the remote registry does provide Retina with the ability to verify if specific security patches are installed.

Retina only uses the credentials of the account under which it is executed when accessing the remote system's registry -- it will not use any "net use" supplied credentials. Therefore, we recommend executing Retina as a domain administrator account.

How To Fix:

Ensure that the system has remote registry capabilities enabled, and that you have administrative rights on the system.

Machine: XXX.XXX.XXX.XXX

OS Detected: No Matches

No More Details Available

Closed Ports: 57167

No More Details Available

Filtered Ports: 8343

No More Details Available

Open Ports: 25

No More Details Available

Ports: XXX.XXX.XXX.XXX

80: WWW-HTTP - World Wide Web HTTP (Hyper Text Transfer Protocol)
Detected Protocol: HTTP
Port State: Open
Version: WEBLOGIC WEBLOGIC TEMPORARY PATCH FOR CR064988
02/05/2002 15:38:23

139: NETBIOS-SSN - NETBIOS Session Service
Port State: Open
Version: NONE

1416: NOVELL-LU6.2 - Novell LU6.2
Port State: Open

1418: TIMBUKTU-SRV2 - Timbuktu Service 2 Port
Port State: Open

1433: MS-SQL-S - Microsoft-SQL-Server
Port State: Open

1521: NCUBE-LM - nCube License Manager
Port State: Open

1526: PDAP-NP - Prospero Data Access Prot non-priv
Port State: Open

1720:
Port State: Open

1801:
Port State: Open

2059:
Port State: Open

2101:
Port State: Open

2103: ZEPHYR-CLT - Zephyr Serv-HM Conncetion
Port State: Open

2105: EKLOGIN - Kerberos (v4) Encrypted RLogin
Port State: Open

2157:
Port State: Open

3181:
Port State: Open

3200:
Port State: Open

3300:
Port State: Open

3389: MS RDP (Remote Desktop Protocol) / Terminal Services
Port State: Open

3399:
Port State: Open

7070: ARCP (nasa.gov)
Port State: Open

8080: Generic - Shared service port / HTTP Alternate

Port State: Open

8091:
Detected Protocol: HTTP
Port State: Open
Version: SIMPLE, SECURE WEB SERVER 1.1

9100: JETDIRECT - HP JetDirect Card
Port State: Open

9999: DISTINCT - distinct
Port State: Open

20200:
Port State: Open

References

Birkholz, Erik Pace. "Special Ops: Host and Network Security for Microsoft, UNIX, and Oracle". Syngress. 2003.

"BS ISO/IEC 17799:2000 (BS 7799-1:2000) Information technology - Code of practice for information security management". BSI Institute. 2000.

CGISecurity Website. http://www.cgisecurity.com/lib/. (2003 Nov 15)

Endler, David & Sutton, Michael. "Web Application Brute Forcing 101". URL: http://www.blackhat.com/presentations/bh-usa-02/endler/bh-us-02-endler-brute.ppt (2003 Nov 15)

Glaser, JD & Shah, Saumil. "Web Hacking Part 1 & 2". URL: http://www.blackhat.com/presentations/win-usa-01/Glaser-Shah/bh-win-01-glaser-shah.ppt. (2003 Nov 15)
http://www.blackhat.com/presentations/win-usa-01/Glaser-Shah/bh-win-01-glaser-shah2.ppt. (2003 Nov 15)

Grossman, Jeremiah. "Challenges of Automated Web Application Scanning". URL: http://www.whitehatsec.com/ppt/WhiteHat_Blackhat_Federal_2003_v1.7.ppt. (2003 Nov 15)

Groves, Dennis & Pennington, Bill. "Web Application Security". URL: http://www.blackhat.com/presentations/bh-usa-02/bh-us-02-groves-webapps.ppt. (2003 Nov 15)

McClure, Stuart & Shah, Saumil & Shah, Shreeraj. Web Hacking: Attacks and Defense. Addison-Wesley Pub Co. 2002.

Naidu, Krishni. "Web Application Checklist". URL: http://www.sans.org/score/checklists/WebApplicationChecklist.pdf. (2003 Oct 30)

OWASP. "Guide to Building Secure Web Applications v1.1.1". URL: http://www.owasp.org/documentation/guide/1.1/index. (2003 Nov 15)

OWASP. "The Ten Most Critical Web Application Vulnerabilities". URL: http://www.owasp.org/documentation/topten. (2003 Nov 15)

Parker, Donn B. "Fighting Computer Crime: A New Framework for Protecting Information". Wiley & Sons. 1998.

Rhoades, David. "7.3 Auditing Web-based Applications". SANS Institute. 2003.

Russel, Ryan & Dubrawsky, Ido & FX. Stealing the Network: How to Own the Box. Syngress. 2003.

Schiffman, Mike. "Hacker's Challenge : Test Your Incident Response Skills Using 20 Scenarios". McGraw-Hill Osborne Media. 2001.

Schiffman, Mike & Penningtonm, Bill & Pollino, David, & O'Donnellm, Adam J. "Hacker's Challenge 2: Test Your Network Security & Forensic Skills". McGraw-Hill Osborne Media. 2002.

Schneier, Bruce. "Beyond Fear: Thinking Sensibly About Security in an Uncertain World". Copernicus Books. 2003.

Shah, Saumil. "Top Ten Web Hacks". URL: http://www.blackhat.com/presentations/bh-asia-02/bh-asia-02-shah.pdf. (2003 Nov 15)

Scambray, Joel & Shema, Mike. Hacking Web Applications Exposed. McGraw-Hill Osborne Media. 2002.

Stoneburner, Gary, Goguen, Alice & Feringa, Alexis. "Risk Management Guide for Information Technology Systems." Special Publication 800-30, National Institute of Standards and Technology, July 2002. URL: http://csrc.nist.gov/publications/sp800-30.pdf (2003 November 14)

"The Standard of Good Practice for Information Security". Information Security Forum. URL: http://www.isfsecuritystandard.com/index_ie.htm (2003 Nov 15)