Chapter 6
Central Services—Web Application Security Requirements

1.0 Main Points

Web applications may allow attackers to access or corrupt confidential government information or interrupt government services if not appropriately designed or operated.

Government ministries use various web applications to provide services. As owners of these applications, ministries are responsible for their security within the parameters set by the Ministry of Central Services (Central Services). In addition to providing the security policy framework for these web applications, Central Services develops and hosts them in a data centre for ministries.

At December 2015, Central Services' overall information technology (IT) security policy framework was consistent with best practices. But it did not have sufficiently comprehensive procedures and guidance to support the development and operation of secure ministry web applications. In addition, it does not require routine analysis of web application vulnerabilities (weaknesses). With the Ministry's cooperation, we tested the security of 18 ministry websites. Most of them were not sufficiently secure.

We made four recommendations to help ensure new ministry web applications are appropriately designed, and existing web applications are kept secure.

Sufficiently comprehensive procedures and guidance would include working with the ministries to promptly identify and address identified web application vulnerabilities classified as higher risk. Comprehensive procedures support an organized and consistent approach to implementing and maintaining security across ministries. This helps minimize the risk of a breach of government information in the web applications, and other applications and data that Central Services hosts in the data centre.

2.0 Introduction

The Ministry of Central Services, as a central agency, coordinates and delivers IT services to 15 government ministries[1] (ministries) and about 10 other agencies. *The Ministry of Central Services Regulations* makes Central Services responsible for developing, implementing, monitoring, and enforcing IT security policies and standards of the Government of Saskatchewan.[2] These IT security policies and standards include those related to the development and operation of web applications owned by ministries and agencies.

In 2014-15, Central Services spent $121.8 million (2013-14: $121.1 million) to provide IT services, of which it recovered $107.5 million (2013-14: $112.8 million).[3] Central Services

[1] This includes the Public Service Commission and Executive Council. Central Services also provides IT services for itself. It does not provide IT services to the Ministry of Health.
[2] *The Ministry of Central Services Regulations*, section 3(k).
[3] Ministry of Central Services financial records and *Annual Report for 2013-14*, pp. 5 and 18.

operates on a cost-recovery basis for IT services; that is, the ministries and agencies reimburse Central Services for the IT services it provides or coordinates on their behalf.

This chapter reports the results of our audit of whether Central Services had security requirements (e.g., policies, standards) that were consistent with best practices for the development and operation of ministry web applications. The **Glossary** in **Section 6.0** defines many of the terms used in this chapter.

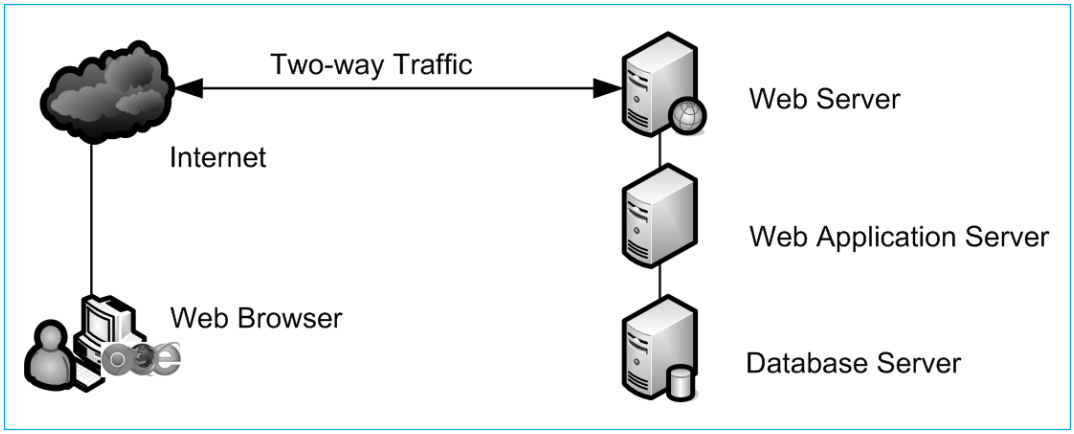
3.0 Background

3.1 What are Web Applications?

Web applications are computer programs that are built into websites, and help websites work. **Figure 1** outlines a typical web application.

The public uses a web browser (e.g., Microsoft Internet Explorer, Mozilla Firefox, Google Chrome) to interact with a website (i.e., provide or obtain information) through the Internet. A web server receives the request for information from the public. It uses software (the web application) to obtain a response from other computer servers (e.g., web application server, database server). The web application then packages and delivers the response back to the user's browser, which displays the web page with the information. For example, web applications are used when filling out a form, creating an account, using a shopping cart, or using the search capability on a website.

Figure 1—Example Web Application



Source: Provincial Auditor Saskatchewan (2016).

3.2 Security Risks of Web Applications

While web applications contribute to the usefulness of the Internet, they can create security problems if not appropriately developed or operated. A 2014 SANS[4] survey reports 29% of organizations experienced one or more security breaches due to an application security vulnerability (i.e., weakness) during the 18 months prior to the

[4] The SANS Institute is a cooperative information security research and education organization.

survey.[5] A 2015 SANS survey reports 74% (2014: 38%) of organizations considered public-facing web applications to be their highest security risk.[6]

Attackers can use weaknesses in web applications to see sensitive information (e.g., credit card, banking, birthdate information) while it is being processed by the web application (i.e., data in transit). Weaknesses in web applications can allow attackers to gain access to data stored by the web applications or other applications in the same network(s). In addition, attackers can exploit weaknesses in web applications to put systems and data belonging to public users at risk. For example, attackers exploited a type of web application weakness called injection flaws[7] at Sony in 2011 and at Bell Canada in 2014 to gain access to customer information including credit card numbers.[8]

A wide variety of tools are designed to exploit potential weaknesses in websites. These tools are available online and many are free. Since websites are on the Internet, attacks can be carried out from anywhere in the world.

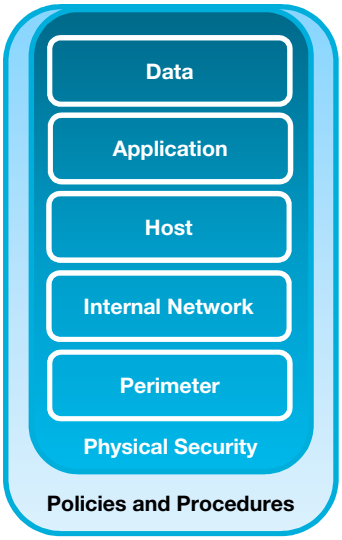
To protect against the many ways that an attacker may attempt to gain access to systems and data, many organizations apply a defense-in-depth strategy as outlined in **Figure 2**. The principle of defense-in-depth is that layered security mechanisms increase security of the system as a whole.

Figure 2—Defense-in-depth

The diagram to the right illustrates the common layers where security may be implemented. If an attack causes one security mechanism to fail, other mechanisms may still provide the necessary security to protect the system. Due to the ongoing discovery of new security weaknesses and that no security mechanism is foolproof, securing only one layer (e.g., perimeter) is not adequate. It is important that security be addressed at all layers based on assessed risk (i.e., defense-in-depth).

For example, a web application system can be protected using security controls applied around the network perimeter and in the computer network (e.g., intrusion detection systems, firewalls). The protection can be improved by securely setting up the servers that the web application and the related programs and databases are hosted on. Security is further improved by following secure coding (i.e., programming) practices and secure deployment to help make sure that the web applications themselves operate securely. The combination of security controls at multiple layers makes it more difficult to attack the web application.

Security policies and procedures are the foundation for the other security layers because they set out the minimum security expectations for people developing and operating the web application to enable an organized and consistent approach to security.



Sources: Based on information from www.owasp.org/index.php/Defense_in_depth (16 March 2016). Diagram from The Business Forum, Antivirus Defense-In-Depth Guide (2015).

[5] SANS, *Survey on Application Security Programs and Practices*, (2014), p. 18. www.sans.org/reading-room/whitepapers/analyst/survey-application-security-programs-practices-34765 (19 April 2016).
[6] SANS, *2015 State of Application Security: Closing the Gap*, (2015), p. 8. https://info.whitehatsec.com/rs/675-YBI-674/images/SANS_Survey_AppSec_2015_WhiteHat.pdf (11 January 2016).
[7] Injection flaws are weaknesses in a poorly-designed web application that allow attackers to gain access by entering code in areas of the website (such as forms) that are used to gather information or receive requests. These attacks can result in data loss or corruption, denial of service, or complete host takeover. If attackers can access the web application's databases, they can potentially attack visitors to the website.
[8] http://business.financialpost.com/fp-tech-desk/bell-hack-attack-that-affected-more-than-20000-customers-shows-rising-security-threat (15 May 2015).

3.3 Central Services' Responsibilities for Security of Web Applications

Central Services has a dual role. First, Central Services is responsible for developing, implementing, monitoring, and enforcing IT security policies and standards for itself, 14 other ministries, and about 10 other agencies.[9] Second, Central Services delivers some IT services to these ministries and agencies (such as developing web applications, and hosting[10] web applications).

The ministries have at least 50 web applications that provide services to and interact with the public.[11] For example, when a user applies for a student loan, pays a fine, or orders high school transcripts, the ministries' web applications allow these activities to occur over the Internet.

The ministries, including Central Services itself, own web applications. As the owners of the applications, the ministries are responsible for their security, but within the context of Central Services' security policies and procedures.

Security policies and procedures provide the foundation for security. Strong policies and procedures set out the security expectations for developing and operating the web application to enable an organized and consistent approach to security.

Strong security policies and procedures are particularly important for web applications in that weaknesses in one web application that Central Services hosts may increase the risk of breaches of other systems and data that Central Services hosts.

As such, security policies and procedures are fundamental for protecting the government information accessible through the ministries' websites. Strong web application security policies and procedures for the ministries reduce the risk of security weaknesses and, in turn, the potential for attackers to access or corrupt confidential government information or disrupt government services.

4.0 Audit Objective, Scope, Criteria, and Conclusion

The objective of this audit was to assess whether, at December 31, 2015, the Ministry of Central Services had security requirements that were consistent with best practices for the development and operation of government ministry web applications.

For the purposes of this audit, security requirements include Central Services' policies, standards, procedures, forms, and other guidance for use by the ministries including by Central Services itself. We considered best practices published by recognized leaders for IT technical security standards for web applications, such as the Open Web Application Security Project (OWASP)[12] and the National Institute of Standards and Technology (NIST).[13]

[9] *The Ministry of Central Services Regulations*, section 3(k).
[10] Hosting is where the IT system servers and data are located at the service provider. Central Services has contracted hosting of ministry servers and data to an external service provider.
[11] Central Services did not have a complete list of ministry web applications at December 31, 2015.
[12] OWASP is an international not-for-profit organization focused on improving the security of software.
[13] NIST is a non-regulatory federal agency within the U.S. Department of Commerce that works with industry to develop and apply technology, measurements, and standards, including in the area of information technology.

To conduct this audit, we examined the various security requirements of Central Services. We compared these requirements for consistency with best practices. We interviewed Central Services staff involved in developing, maintaining, implementing, and monitoring requirements for web application security. After obtaining Central Services' permission, we tested the security of a selection of websites related to the ministries' web applications to assess if they demonstrated consistency with best practices. To facilitate our testing, Central Services' security staff did not respond to alerts indicating external scanning of web applications was occurring. We did not attempt to exploit the weaknesses we identified (i.e., did not try to gain access to the applications and data).

We followed the standards for assurance engagements published in the *CPA Canada Handbook – Assurance*. To evaluate Central Services' security requirements, we used criteria based on our related work, reviews of literature including reports of other auditors, and consultations with management. Central Services' management agreed with these criteria (see **Figure 3**).

Figure 3—Audit Criteria

1. Support overall security objectives
1.1 Align with overall security direction (i.e., high-level policy)
1.2 Describe the intent for each security requirement
2. Align with leading technical security standards for web applications
2.1 Reference security framework(s) used to develop the requirements
2.2 Reflect a risk-based approach
2.3 Review periodically
3. Clearly outline roles and responsibilities
3.1 Align with legislative responsibilities
3.2 Address necessary stages of development and operation
3.3 Provide sufficient detail to support implementation
4. Require verification that security objectives are met
4.1 Require reporting of verification activities and results
4.2 Define processes for analyzing progress towards security objectives
4.3 Define processes for resolving security concerns

While the Ministry of Central Services had an overall security policy framework consistent with best practices, we concluded that, at December 31, 2015, it did not have sufficiently comprehensive procedures and guidance to support the development and operation of secure government ministry web applications. Sufficiently comprehensive procedures and guidance would include working with the ministries to promptly identify and address identified web application vulnerabilities classified as higher risk.

5.0 Key Findings and Recommendations

In this section, we set out our key findings along with related recommendations.

5.1 Information Security Policy Provides Adequate Framework

Security best practices expect organizations to have an overall security framework.[14]

[14] *NIST Special Publication 800-100: Information Security Handbook: A Guide for Managers*, (2006), p. 14.

At December 2015, Central Services had a new Information Security Policy (i.e., the Government of Saskatchewan Information Security Policy [Security Policy]) that it used to communicate its high-level security requirements to its staff (e.g., application developers) and to the ministries' staff (i.e., application owners). In October 2015, Central Services approved this Security Policy. This Security Policy provides Central Services with an overall security framework.

This new Security Policy consolidated and replaced the previous security-related policies (e.g., the Information Technology Office Security Policy (2004), Government of Saskatchewan 2010 Security Standards). Central Services based this new Security Policy on an industry-accepted IT security framework (i.e., best practice) – ISO/IEC 27002:2013.[15] The new Security Policy applies to all types of IT systems owned and operated by government ministries (including itself) and about 10 agencies. While this Security Policy is not just for the security of web applications, it provides a security framework for them.

5.2 Key Information on Web Applications Needed

Security best practices expect organizations to maintain key information about applications that they host or own (e.g., business purpose, application/information owner, risk classification, software version, and servers they are running on).[16]

We found that Central Services did not have complete information on the nature and extent of web applications that it hosts on behalf of the ministries. It had begun to develop a list of web applications that included the following: the name of the ministry that owned the web application, the business purpose, the web address (i.e., URL), and whether a password was required to access the website. However, at December 2015, the list did not include all ministry web applications that Central Services hosted and that are subject to its Security Policy, nor all key details about the web applications (e.g., risk classification, software version, server the applications runs on). Key information about applications (including web applications) would help ensure Central Services designs procedures and guidelines supporting the Security Policy that address risks significant to the ministries.

1. **We recommend that the Ministry of Central Services document key information about all ministry web applications that are subject to its security policy.**

5.3 Security-focused Procedures Needed

Security best practices expect organizations to have detailed procedures and supports to assist staff in implementing the overall security policy framework when managing applications (such as web applications).[17] These detailed procedures would help users

[15] ISO/IEC 27002:2013 gives guidance for organizational information security standards and information security management practices including the selection, implementation and management of controls taking into consideration an organization's information security environment.
[16] Open Web Application Security Project. *OWASP Top 10 – 2013: The Ten Most Critical Web Application Security Risks*, (2014).
[17] ISACA, *An Introduction to the Business Model for Information Security*, (2009), p. 5.

interpret and implement the expectations set out in the overall security policy on a consistent basis. Use of consistent and standardized processes creates efficiencies within and across the ministries. It also helps ensure security decisions for one application do not negatively affect security of the rest of the network. Also, because IT best practices for security offer alternate approaches to managing security, the detailed procedures would either explicitly set expected processes, or alternately which specific IT best practices they expect staff to follow in various situations (e.g., when developing web applications, monitoring web applications). Some organizations develop their own detailed guidance instead of referring to specific leading technical web application security standards for staff to use when assessing and responding to risks to their web applications.

Figure 4 provides examples of alternative leading technical security standards for web applications. One option (OWASP) sets information about how to address critical web application security risks including verifying controls, while another option (NIST) sets guidelines such as securing web services. A third option (Control Objectives for Information and Related Technology or COBIT) sets a governance and management framework for IT. Explicitly setting expected processes or identifying which specific IT best practices to follow facilitates having an organized approach to security across multiple organizations (e.g., the ministries).

Figure 4—Examples of Alternate Technical Security Standards to Support Web Application Procedures and Guidelines

OWASP's *Application Security Verification Standard* (2014) includes detailed considerations related to: access control, management of a session between a user and a web application, malicious input that may be entered into a web application, and data protection. OWASP provides checklists to help determine controls are in place in these areas (e.g., sessions timeout after a specified period of inactivity, controls prevent SQL injection, purge or invalidate temporary copies of sensitive data once no longer needed).

NIST Special Publication 800-95: Guide to Secure Web Services includes detailed considerations related to: web service security functions and related technologies (e.g., authentication, identity management, accountability throughout a service chain, availability of web services); web portals for users (e.g., to control user authorization and access); security of web services for legacy applications; and uses of security tools and technologies (e.g., for development and testing).

COBIT 5 provides a detailed framework for IT governance and management of enterprise IT, which encompasses web application security. *COBIT and Application Controls: A Management Guide* includes detailed guidance for designing and operating application controls, including interdependency with other controls and responsibilities of business and IT units.

Source: Based on information from the technical security standards referenced above, (2016).

At December 2015, Central Services' Security Policy had limited supporting procedures and guidance to help staff from the ministries, including its own staff, interpret and implement sections of its overall Security Policy that relate to web applications.[18]

Central Services had guidance that set out expected processes for the following areas: changes to applications, incident and problem management (i.e., identifying and correcting issues resulting from a specific situation and analyzing the event to prevent a future reoccurrence), and approving exceptions to the Security Policy (e.g., allowing a shorter password length for an application). It also set standard technologies that Central Services will support (e.g., web application languages: Java, Microsoft, .NET).

In addition, it has a number of template forms (see examples in **Figure 5**) that it expected staff to complete for a number of key areas. The purpose of these templates is

[18] Central Services' previous security-related policies also had limited supporting procedures and guidance related to web applications.

to engage Central Services staff to provide guidance to the ministries. The templates do not provide detailed explanations to help staff understand and correctly assess the information required.

Figure 5—Examples of Security-related Templates Available at December 2015

High Level Solution Design Template: A central document that describes in both developer's terms and customer's terms the business, application and information, and technical (hardware and software) infrastructure requirements for implementing the requirements for a project.

Project Security Compliance Assessment Template: Summarizes risk assessment work completed by Central Services and findings (e.g., risks identified) for a ministry project (e.g., development of a web application).

Technical Security Assessment Template: Summarizes risk assessment work completed by Central Services with the aid of IT tools and resulting findings (e.g., risks identified) for a ministry web application.

Risk Management Decision Item Template: Documents a request for exemption from Central Services' security policies, the related risk assessment, and the resulting risk decision (i.e., to accept or try to mitigate the risk).

Source: Based on Central Services security-related templates, (2016).

Overall, although Central Services set out security requirements in its Security Policy, it did not set out supporting procedures and guidelines in many key areas related to IT security for web applications. **Figure 6** provides examples where it set limited or no procedures and guidelines for key sections in its Security Policy related to web applications.

Figure 6—Examples of Key Areas with Limited or No Security Procedures or Guidelines at December 2015

Central Services set limited or no procedures and guidelines for the following key sections in its Security Policy that relate to web applications *(related Security Policy section number)*:

- Maintaining an inventory of all important assets associated with information systems (e.g., web applications) including documentation of the assigned information and service owners (*sections 4.1.1 and 4.1.2*)
- Using a key management system to support the use of cryptographic controls (*section 6.1.2*)
- Regularly assessing information system vulnerabilities and the management of associated risk (*section 8.6.1*)
- Identifying security controls as part of business requirements for new information systems or enhancements to existing information systems (e.g., development methodology and secure coding guidelines) (*section 10.1.1*)
- Establishing, documenting, maintaining, and applying secure information system engineering principles (*section 10.2.5*)
- Maintaining documentation of the statutory, regulatory, and contractual requirements for each information system (*section 14.1.1*)
- Regularly reviewing information systems for compliance with security policies and standards (*section 14.2.3*)

Source: Based on analysis of the Government of Saskatchewan Information Security Policy, (2016).

When it comes to information technology, change is a constant; this is particularly true as it relates to the adoption and use of web applications. Significant threats to existing web applications emerge as attackers become more sophisticated in exploiting weaknesses.[19] As such, it is important that IT security policies are kept current and supporting procedures and guidelines are dynamic and responsive to changes to IT security risks.

[19] Open Web Application Security Project, *OWASP Top 10 – 2103: The Ten Most Critical Web Application Security Risks*, (2014).

Central Services intends to revisit its Security Policy at least every two years. It had not set out how often it planned to update its procedures and guidelines.

Comprehensive security-related procedures and guidelines specify how to achieve the Security Policy, and provide clear direction on who is responsible for making sure the procedures are followed. They allow for consistent treatment of similar situations across the ministries. Without comprehensive procedures and guidelines for web applications, Central Services increases the risk of the ministries' staff, including its own staff, not fully understanding its Security Policy, and not implementing it properly. Not properly implementing its Security Policy increases the risk of security not being maintained to an appropriate level thereby increasing the likelihood of security breaches.

2. We recommend that the Ministry of Central Services develop and maintain comprehensive procedures and guidelines to support the development and operation of secure web applications.

5.4 Web Application Development Considers Security

Security best practices expect developers to incorporate security into the design of the web applications to reduce security risks to levels acceptable to the owner of the application with consideration of security implications on others whose applications reside on the same IT network. They also expect developers to confirm that security features embedded in the web application design work as intended.[20]

As noted in **Figure 2**, security policies and procedures are one security mechanism used to increase security of a system as a whole. Other mechanisms include physical security of the system (e.g., restricting access to computers through use of locked doors), perimeter security (e.g., use of firewalls and intrusion detection systems), and internal network security (e.g., securely setting up servers). As reported in our *2016 Report – Volume 1*, Chapter 5, while Central Services appropriately used many of these mechanisms, it had weaknesses in its internal network and perimeter security.

For new ministry web applications, Central Services expected its staff, when developing[21] new web applications for the ministries, to implement safeguards into the application consistent with the high-level expectations included in its Security Policy. We found Central Services did not have procedures to ensure web developers had access to written updates about evolving security weaknesses identified by industry, so they could consider these when developing new web applications.

Also, as noted in **Section 5.3**, Central Services did not set detailed procedures related to developing websites. Detailed procedures and guidance would help to ensure an organized and consistent basis for making security decisions about web applications.

We found, consistent with best practices and its Security Policy, Central Services required security testing during the development stage of web applications to confirm security worked as intended.

[20] ISACA, *Web Application Security: Business and Risk Considerations*, (2011), p. 10.
[21] Central Services has a direct role in developing new ministry web applications either by having its staff develop the application, or using contractors for the development.

5.5 Proactive Routine Monitoring of Compliance with IT Security Policies Needed

Security best practices also expect processes for ongoing monitoring of the security of the applications.[22] For example, routine use of security vulnerability testing for a web application can identify vulnerabilities found since an application was first developed.

Routine security vulnerability testing helps organizations know the significance and magnitude of security risks facing their systems and data. Security vulnerability testing is designed to identify weaknesses, and classify the risks of those weaknesses from low to high. See **Figure 7** for description of risk classes. Organizations can then use this information to identify and take corrective actions.

Figure 7—Risk Classification of Security Vulnerabilities

A web application security vulnerability can be classified as high, medium, or low risk, as follows:

High:	A vulnerability that could let an attacker execute commands on the server, retrieve and modify confidential information, or view source code, system files, and sensitive error messages.
Medium:	Other errors or issues that could be sensitive (e.g., let an attacker gather sensitive information about the web application such as machine name and/or sensitive file locations).
Low:	Interesting issues, or issues that could evolve into a higher risk vulnerability.

Source: Adapted from Auditor General of Alberta, *Report of the Auditor General of Alberta—October 2008,* p. 58.

Central Services, as legislated, has responsibility for developing, implementing, monitoring, and enforcing the overall security policy and standards related to the ministries' IT systems and data including web applications.

Through its Security Policy and its agreements with individual ministries, it requires the ministries (owners of applications) to comply with its policies and keep their applications and data secure. In addition, through its Security Policy, because it hosts the ministries' systems, it expects its staff to periodically use IT tools to identify security weaknesses. However, Central Services has not set out the nature and extent of tests it expects (e.g., what types of tests and how often).

We found Central Services does not complete routine testing of web application vulnerabilities, or require the ministries to do so. Rather Central Services may carry out or contract for these tests only upon request of the ministries. If the ministries request such tests using contractors, Central Services requires the ministries to share vulnerabilities identified.

Instead of a proactive approach to routinely testing the adequacy of the security of the ministries' web applications, Central Services used a reactive approach. As noted in **Section 5.3**, it had incident and problem management processes. It required the ministries to report security incidences (e.g., security breach), and worked with the ministries to resolve the situations. It tracked information about incidents and vulnerabilities reported and resolved.

We selected 18 websites related to existing ministry web applications. These included web applications of various types (e.g., those that process financial information or house

[22] ISACA, *Web Application Security: Business and Risk Considerations*, (2011), pp. 10 and 12.

confidential personal information). These web applications were owned by various ministries (11 different ministries), and were of various ages (both newer and older).

In our test of these websites, we identified over 1,400 vulnerabilities with risks that varied in criticality from low to high. One corrective measure may address multiple vulnerabilities identified, such that the number of vulnerabilities identified does not necessarily equate to the level of effort required to correct them. For the 18 ministry websites we tested, we found:

- The number and risk classification of the vulnerabilities varied by website (e.g., eight websites had three or fewer medium- and high-risk vulnerabilities, while one website[23] had over 100 high-risk vulnerabilities); overall, 22% of the vulnerabilities identified were classified as medium and/or high risk. Also, as described in **Figure 8**, we identified a well-known high-risk weakness affecting 10 of the 18 ministry websites. Best practices expect organizations to be aware of new significant vulnerabilities and take steps to proactively fix them.[24]

Figure 8—Example of a Security Vulnerability Affecting Many Ministries

For 10 of the 18 websites we tested, we identified a significant encryption security vulnerability that has been well known in the IT community since the early 2010s. The IT community quickly developed ways (e.g., recommended updates to software) to fix the vulnerability. If exploited, this encryption vulnerability allows an attacker to access data from an encrypted session. This data could include passwords, cookies, and other authentication tokens that may be used to gain access to the website (e.g., impersonate the user, access the database).

Source: Based on testing and research by Provincial Auditor Saskatchewan (2015).

- Weaknesses were wide-spread across the ministries in that they were identified in 17 of the 18 ministry websites we tested

- Weaknesses were found in both newer (e.g., 2013) and older (e.g., 2000) websites

Overall, most of websites we tested were not sufficiently secure (that is, were at medium to high risk of letting an attacker gain access or gather sensitive information about the web application).

Not using routine testing of the security of ministry web applications increases the risk that higher-risk vulnerabilities are not identified, and addressed before security breaches occur. This in turn increases the risk that ministry web applications can be compromised, and sensitive data lost or accessed.

3. We recommend that the Ministry of Central Services require routine analysis of web application vulnerabilities to monitor compliance with its security policy.

We shared the detailed results of testing of web applications with Central Services to enable it to work with the related ministries to identify and take corrective actions. As previously noted, one corrective measure may address multiple vulnerabilities identified.

[23] Management advised us that work has begun related to this website that will address existing vulnerabilities.
[24] www.owasp.org/index.php/Virtual_Patching_Best_Practices (14 March 2016).

Not taking timely corrective action on higher-risk vulnerabilities makes it easier for ministry web applications to be breached.

4. **We recommend that the Ministry of Central Services work with the ministries to address identified higher-risk web application vulnerabilities.**

6.0 Glossary

Application – A software program. This includes programs such as word processors, spreadsheets, database programs, accounting programs, etc.

Database – A comprehensive collection of related data organized for convenient access in a computer.

Change Management – An organized approach for introducing changes into a program or process, used to minimize unintended consequences.

Data Centre – A central location for computer network hardware and software, especially storage devices for data.

Defense-in-depth – The practice of using layered security mechanisms to increase security of the system as a whole. If an attack causes one security mechanism to fail, other mechanisms may still provide the necessary security to protect the system.

Firewall – Software and/or hardware intended to restrict or block access to a network or computer. Firewalls can be set up using firewall rules to only allow certain types of data through.

Intrusion detection system – Software and/or hardware designed to detect a security breach by identifying inappropriate access or changes taking place within a computer or network.

Network – A group of computers that communicate with each other.

Patch – An update to a computer program or system designed to fix a known problem or vulnerability.

Procedure – An established or official way of doing something.

Secure Coding – The practice of developing application software in a way that reduces the risk of accidental introduction of security vulnerabilities in software before it is deployed.

Security Vulnerability – An unintended weakness in a computer system that exposes it to the potential exploitation such as unauthorized access or malware (e.g., viruses).

Server – A computer that hosts systems or data for use by other computers on a network.

Software – A set of machine-readable instructions that directs a computer to perform specific operations.

Web Browser – A software program used by a computer to locate, retrieve, and display information from a website (e.g., Microsoft Internet Explorer, Mozilla Firefox, and Google Chrome).

7.0 Selected References

Auditor General of Alberta. (2008). *Report of the Auditor General of Alberta – October 2008, Protecting Information Assets.* Edmonton: Author.

ISACA. (2011). *Web Application Security: Business and Risk Considerations*. www.isaca.org/knowledge-center/research/researchdeliverables/pages/web-application-security-business-and-risk-considerations.aspx (14 March 2016).

ISACA. (2009). *An Introduction to the Business Model for Information Security*. www.isaca.org/knowledge-center/bmis/documents/introtobmis.pdf (14 March 2016).

National Institute of Standards and Technology. (2007). *Guidelines on Securing Public Web Servers*. http://csrc.nist.gov/publications/nistpubs/800-44-ver2/SP800-44v2.pdf (20 May 2015).

National Institute of Standards and Technology. (2007). *Guide to Secure Web Services*. http://csrc.nist.gov/publications/nistpubs/800-95/SP800-95.pdf (14 March 2016)

Office of the Auditor General of British Columbia. (2014). *Information Technology Compendium, Web Application Security Audit*. Victoria: Author.

Open Web Application Security Project. (2014). *Application Security Verification Standard*. www.owasp.org/images/5/58/OWASP_ASVS_Version_2.pdf (15 May 2015).

Open Web Application Security Project. (2014). *OWASP Top 10 – 2013: The Ten Most Critical Web Application Security Risks*. www.owasp.org/index.php/Category:OWASP_Top_Ten_Project (14 March 2016).

Provincial Auditor of Saskatchewan. (2014). *2014 Report – Volume 1, Chapter 7, Central Services-Information Technology Division-Data Centre.* Regina: Author.

The Information Systems Audit and Control Association. (2012). *COBIT 5*. Rolling Meadows: Author.