City of Cambridge
Executive Department

YI-AN HUANG
City Manager

City of Cambridge Written Information Security Policy ("WISP")

I. Policy Overview and Authorization

1. Policy Overview

The City of Cambridge (the "City") has adopted and continually monitors and enhances a comprehensive Written Information Security Policy (the "WISP" or the "Policy") in order to develop, introduce and maintain appropriate and effective administrative, physical, and technical safeguards for the protection of Personal Information ("PI") (defined below), also known as Personally Identifiable Information ("PII") (collectively hereinafter "PI"). The WISP is a component of the City's cybersecurity strategy which also includes a comprehensive, industry-recognized cybersecurity framework intended to safeguard all information deemed sensitive by the federal government, the Commonwealth, and the City, including PI,.

This Policy is intended to safeguard the PI of any Massachusetts resident which the City obtains or which any third party service providers, independent contractors, consultants, and subcontractors and their agents and employees (collectively "Contractors") of the City obtains on behalf of the City, and to document the City's policies for collecting, storing, accessing, using, transmitting, and protecting electronic, paper, and other records containing the aforementioned PI in general conformity with the obligations set forth in the regulations promulgated by the Commonwealth's Office of Consumer Affairs and Business Regulations, 201 CMR 17.00 - Standards for the Protection of Personal Information of Residents of the Commonwealth (the "Regulations") pursuant to the authority outlined in G. L. c. 93H, § 2. Although the Regulations do not require the City to adopt a WISP (i.e., the definition of person in 201 CMR 17.02 specifically excludes municipalities), the City has deemed it a best practice to adopt a WISP. All City employees who handle PI as defined in this Policy shall be given a copy of this Policy and shall be required to review it and submit an acknowledgement to the Personnel Department, upon commencement of their employment with the City for new employees or prior to handling PI for existing employees, stating that they have reviewed this Policy. Existing employees already handling PI shall have thirty days from the date of issuance of this Policy to submit their acknowledgement to the Personnel Department.

The Cambridge Fire Department, Cambridge Police Department, and the Emergency Communications Department (collectively, "Public Safety") and any staff in other City departments supporting a Public Safety department's functions shall, in addition to complying with this Policy, abide by the United States Department of Justice's "Criminal Justice Information

System ("CJIS") Security Policy," as it may be amended form time to time and any applicable CJIS User Agreement. To the extent there is a conflict between the provisions of this Policy and the CJIS Security Policy, the CJIS Security Policy shall prevail with respect to Public Safety operations.

The School Department shall be governed by the WISP issued by the School Department.

2. Personal Information Definition ("PI")

For purposes of the WISP, Personal Information ("PI") means a resident of the Commonwealth's first name and last name, or first initial and last name, in combination with any one or more of the following data elements that relate to such person:

a) Social Security Number ("SSN");

b) driver's license number or state-issued identification card number; or

c) financial account number, or credit or debit card number, with or without any required security code, access code, personal identification number or password, that would permit access to a person's financial account.

The definition of PI does not include information that may be obtained from publicly available information, or from federal, state or local government records lawfully made available to the general public.

3. Policy Approval

The Policy has been reviewed, approved and adopted by the City Manager, and may only be amended, suspended, or terminated by the City Manager.

II. Policy Purpose, Scope and Administration

1. Purpose and Scope

The purpose of the Policy is to establish administrative, technical, and physical safeguards to protect PI that is owned, licensed, stored, or maintained by the City, whether such PI is contained in paper or electronic records, or in any other form.

The Policy is designed to promote the security and confidentiality of PI, to anticipate and protect against threats or risks to the security or integrity of PI, and to safeguard against unauthorized internal and external access to acquire or misuse PI that could lead to identity theft or fraud.

2. WISP Administration

A. Policy Administration. The City's Chief Information Officer and/or their designee(s) in the Information Technology Department shall be the "WISP Coordinator" for this Policy.

B. Responsibilities of Information Security Policy Coordinator. The WISP Coordinator will be responsible for performing each of the following responsibilities, among others:

i. Develop, implement, administer, monitor, review, and update this Policy, in coordination with the Personnel Director and the City Solicitor and approval of the City Manager, from time to time;

ii. Oversee ongoing employee training and any communications involving the Policy;

iii. Address any information security issues, including employee compliance and access to PI by former employees, that may arise from time to time, and provide input to management and the Personnel Department regarding the imposition of disciplinary measures for violations of the Policy;

iv. Take all reasonable steps to verify that any Contractor with access to PI have the capacity to protect such PI in the manner consistent with this Policy, and that any such Contractor applies protective security measures at least as stringent as those required by the Policy; and

v. As part of the submission of responses to Invitations for Bids, Requests for Proposals, or Requests for Quotes, all prospective City contractors for those contracts which will involve the handling of PI shall be required to acknowledge receipt of this Policy inclusive of its exhibits and submit an acknowledgement to the City agreeing to provide a copy of this Policy to any employee of that contractor who will have access to PI as part of their work pursuant to the contract. For those contracts which will involve the handling of PI that do not require a bid, proposal or quote, the Department Head for each department shall be responsible to ensure that the lead contact for any such contractor receives this Policy and submits an acknowledgment agreeing to provide a copy of this Policy to any employee of that contractor who will handle PI pursuant to any work under the contract prior to commencing any work pursuant to that contract. In the event that a contractor has stricter information security requirements than those outlined in this Policy and complying with such stricter provisions would present a conflict with the provisions of this Policy, such a contractor shall submit to the WISP Coordinator for review and approval a statement explaining how that contractor's information security protocols will achieve each of the requirements of this Policy. The WISP Coordinator shall be responsible to work with City departments to ensure compliance with the above requirements in this Subsection II(2)(B)(v).

III. Policy Compliance

1. Compliance with Policy

A. **Compliance**. All employees (full-time, part-time, or temporary) and Contractors are

subject to the applicable requirements outlined in the Policy.

B. **Non-Compliance**. Instances of non-compliance with the Policy must be reported immediately to the WISP Coordinator. Violations may result in disciplinary action by the City, up to and including termination of employment or, where a Contractor is involved, termination of contracts.

C. **Non-Retaliation**. It is against City policy to retaliate against a person who reports a violation of the Policy or who cooperates in an investigation regarding non-compliance with the Policy. Any such retaliation will result in disciplinary action by the City, up to and including termination of employment or, where a Contractor is involved, termination of contracts.

IV. WISP General Governance

1. Risk Assessment and Policy Review

A. **Personal Information Inventorying**. The City will periodically evaluate its holding of electronic, paper and other records, electronic systems, and storage media (including laptops and portable devices used to store PI) to determine where PI is located.

B. **Risk Assessment**. As part of a wider and more comprehensive cybersecurity assessment, the City will periodically:

i. Conduct a review to identify reasonably foreseeable internal and external risks to the security, confidentiality, or integrity of any electronic, paper, or other records containing PI;

ii. Assess the likelihood and potential damage of these threats, taking into consideration the sensitivity of the PI;

iii. Evaluate the sufficiency of this Policy to control those risks; and

iv. Revise the Policy to minimize the risks.

The risk assessment will include, but may not be limited to, an assessment of internal and external risks associated with ongoing employee training, employee and Contractor compliance with the Policy, and the means for detecting and preventing security system failures.

C. **Periodic Policy Review**. The WISP Coordinator will conduct a review of the Policy at least annually, and whenever there is a material change in the City's business or IT practices that may reasonably involve the security or integrity of records or files containing PI.

2. Security Awareness

A. **Training**. The City will provide mandatory education and training regarding Information Security to all employees, including Contractors' employees, who will have access to PI as a requirement of their employment or contract with the City. All employees or employees of Contractors who manage PI databases shall review the City's Vendor/Cloud/Hosted Software as a Service Assessment attached hereto as

Exhibit A and incorporated herein by reference, and the City's Web Security Standards and Practices is attached hereto as **Exhibit B** and incorporated herein by reference.

B. **Third Party Service Providers, Independent Contractors, and Consultants, Subcontractors**. The City will communicate relevant policies and procedures under this Policy to its Contractors who may have access to PI through their contracts and/or services provided to the City.

3. Third-Party Service Providers

A. **Vetting Process**. Before engaging a Contractor who will have access to PI, the City will conduct reasonable due diligence to assess whether the prospective Contractor is capable of safeguarding PI in the manner required by the Policy. Due diligence efforts may include, but are not necessarily limited to, discussions with the prospective Contractor's personnel, reviewing the prospective Contractor's privacy and/or information security policies, and/or requesting the prospective Contractor to complete a security questionnaire or otherwise answer security-related questions. The City may also enter into contractual agreements with its Contractors to protect PI disclosed to such Contractors by the City. Each employee of a Contractor who will have access to PI shall be given a copy of the City's Vendor/Cloud/Hosted Software as a Service Assessment, a copy of which is attached hereto as **Exhibit A** and is incorporated herein by reference, and shall acknowledge that they have received and read that document prior to commencing work under any agreement with the City pursuant to which they will have access to PI.

B. **Monitoring.** The City will periodically monitor and review the performance of its Contractors who have access to City systems and/or PI in order to ensure that each such Contractor is applying protective security measures at least as stringent as those required by this Policy.

4. Incident Management

A. **Reporting Obligation**. Employees and Contractors are required to immediately report to the WISP Coordinator any security violations, breaches of security, or suspicious or unauthorized use of PI contained in records or files belonging to the City (collectively "Security Incident").

B. **Incident Review**. The City will document any responsive actions taken in connection with each Security Incident. The City will conduct a prompt review of all Security Incidents, including Security Incidents where the law requires notification, and determine whether any changes to the Policy are required to improve the security of records and files containing PI.

V. PI Collection and Handling

1. PI Collection, Retention and Return or Disposal

A. **Collection and Retention of PI**. The City collects and maintains records and files containing PI of the type, and for the length of time, reasonably necessary to meet the City's legitimate business requirements, or as otherwise necessary for City to comply

with federal, state, or local laws or requirements. The City will periodically review its records, files, and form documentation to ensure that the City is not collecting and retaining PI unless there is a compelling business need to do so or unless retention is required by law.

B. **Return or Disposal of PI**. All employees and Contractors of City are required upon termination or resignation from the City (for any reason), or upon the request of the City or the WISP Coordinator, to return all records and files containing PI obtained during the course of carrying out their duties as employees of the City or a City contractor , in any form that may at the time of such termination be in their possession, custody or control, including PI stored on laptops, portable devices (such as City or personal cell phones, tablets, thumb drives, CDs, DVDs etc.), or any other media, or in files, records, notes, or papers, and to delete or destroy duplicates of such files. The City Departments, including Personnel and Purchasing Departments, shall incorporate in any employee exit process or Contractor termination process the return of any and all PI to which the employee or Contractor had access, **and shall work with the IT Department to ensure that all usernames and passwords by which that employee or Contractor had access to City databases are permanently disabled.** Destruction of non-duplicate files shall comport with the applicable provisions of the Retention Schedule.

2. PI Handling

PI must be created, stored, disclosed, transmitted, and disposed of in the following manner:

A. **Creation**. When practicable, upon creation of electronic or hard copy documents and files that contain PI, such documents and files must be classified and indicated as "Confidential." A receptacle housing documents containing PI may be marked on the outside, e.g., on the outside of a box or file cabinet, in lieu of stamping every document within the receptacle.

B. **Storage.** Storage of electronic PI should be kept to a minimum, and any PI stored electronically (including on mobile devices) must be encrypted. (Questions regarding City encryption technology should be directed to the IT Department.) Paper documents containing PI must be stored in a locked or otherwise secured desk, file cabinet, office, or an area with controlled access when unattended.

C. **PI Access, Sharing, and Disclosure**. Access to, sharing, and disclosure of records or files containing PI shall be restricted to those persons (employees or Contractors) who are required to have access to such PI in order to accomplish the City's legitimate business purposes or to enable the City to comply with federal, state or local statutory regulations or requirements.

D. **Electronic Transmission**. Electronic transmission of PI must require the prior physical encryption of data beforehand, or transmitted using protocols that have built-in encryption (such as Transport Layer Security "TLS" e-mail of Hypertext Transfer Protocol Secure ("HTTPS") internet connectivity) and must likewise be done with reasonable precaution to ensure the security of such information and to prevent unauthorized interception or disclosure. Voice communications involving PI must also be kept to a minimum and shall not be held in a public forum but performed in a location secured from eavesdropping and unauthorized recording.

E. **Transport**. Physical transport of PI from City facilities (including third party hosting locations), must be done with reasonable precaution and in accordance with any applicable City procedures and/or regulations to ensure the security of such PI and to prevent unauthorized interception or disclosure.

F. **Disposal.** When no longer required, PI must be securely disposed of by the City in accordance with the provisions of the Massachusetts Municipal Records Retention Schedule (the "Retention Schedule"). Paper documents and other hard copies of records or files containing PI determined by City to be no longer needed should be disposed of by cross-cut shredding where available, or otherwise by strip-cut shredding, or by incineration, pulping, redaction, or burning, where destruction is permitted by the Retention Schedule, so that PI cannot be read or reconstructed. Electronic PI determined by the City to no longer be needed must be physically destroyed or securely erased so that PI cannot be accessed, read or reconstructed.

VI. PI Physical and Environmental Security

1. Physical and Environmental Security

A. **Visitors.** Visitors to City buildings are expressly prohibited from accessing any City records or files that contain PI unless the individual is entitled to view the information by law and is given authorization to view such information by the City Manager or his/her designee. In such a case, the visitor shall be monitored at all times by a staff member in the department where the PI is housed to ensure that they only have access to the PI they are authorized to see and that no other information is viewed or accessed.

B. **Physical Security**. The City prohibits physical access to records and files containing PI by any individual who lacks appropriate authorization to access such records by the following means:

i. Limiting access to areas of City buildings where files containing PI are kept to employees and authorized visitors (see Section VII.C. below);

ii. Storing physical files containing PI in locked file cabinets or an area with controlled access when unattended;

iii. Controlling and restricting access to locked file cabinets containing PI to specific key employees

City employees and Contractors are required, upon termination or resignation for any reason, or earlier if requested by the City or the WISP Coordinator, to surrender all keys, IDs, access codes, badges, etc. that facilitate access to City facilities or third party locations that host City systems or to records of the City that may contain PI.

C. **Clean Desk Requirement.** Employees and Contractors of the City must not keep open documents or files containing PI on their desks when they are not at their desks or in any other unsecured, unattended place such as a photocopying or fax station. This Policy applies to both hard copies and electronic copies of records and files containing PI. At the end of the workday, all files and other records containing PI must be secured in a manner that is consistent with this Policy.

VII. PI Technology Security

1. Logical Access Security

A. The City has in place secure user authentication protocols, including the following: (i) control of user IDs and other identifiers; (ii) a reasonably secure method of assigning and selecting passwords; and (iii) control of data security passwords to ensure that such passwords are kept in a location and/or format that does not compromise the security of the data they protect.

B. The City will assign unique identifications plus passwords that are designed to maintain the integrity of the security of the access controls and prohibit the use of Contractor supplied default passwords to each authorized active user.

C. The City will restrict access to City systems and data to authorized users and active user accounts only. Restrictions will further limit access to records and files containing PI to users that have a specific need to access PI in order to perform their job duties. The WISP Coordinator will review in consultation with the City Manager which persons shall be authorized users with an active user account of all City systems and which users need PI to perform their job duties.

D. The City will require that current computer and network passwords are changed periodically. The City will also block access to users after multiple unsuccessful attempts to gain electronic access to records or files containing PI.

E. The City will block unauthorized electronic access to PI by former employees, other former service providers of the City, and other individuals who are no longer authorized users with an active user account.

F. The City will promptly terminate and prohibits electronic access by former employees, former Contractors of the City, and other individuals who are no longer authorized users with an active user account to records and files containing PI. Voicemail access, e-mail access, City internet access, and passwords will also be promptly disabled or blocked.

2. Network Security

A. The City and/or its Contractor will monitor all computer systems for unauthorized use of or access to records and files containing PI.

B. The City and/or its Contractor will continue to maintain and keep updated with authorized security patches, firewall, server operating systems, and file and databases used to process or store PI.

C. The City and/or its Contractor will maintain current versions of system security agent software on all pertinent City servers and devices and regularly distribute and install on City devices up-to-date virus and malware definitions updates.

D. The City and/or its Contractor will periodically conduct internal network vulnerability scans and take prompt action to remediate identified security vulnerabilities.

E. The City and/or its Contractor will periodically conduct exploitation/penetration testing of all City internet facing portals and gateways and take prompt action to remediate security exposures.

3. Portable Device Security

A. The City will require that all portable/mobile devices containing PI be secured with password access.

B. The City will require encryption of all PI stored on laptops or other portable devices.

C. To the extent technically feasible, the City will encrypt all records and files of the City containing PI transmitted across public networks or wirelessly.

Yi-An Huang
Yi-An Huang (Nov 26, 2024 12:19 EST)

Yi-An Huang_____ Dated:11/26/24_____
City Manager

EXHIBIT A: VENDOR/CLOUD/HOSTED SOFTWARE AS A SERVICE ASSESSMENT

City of Cambridge's
Vendor/Cloud/Hosted Software as a Service Assessment

Objective and Scope

The purpose of this document is to provide guidance to assess and evaluate the proposed solution's security and other features and determine key risks.

1. Definitions

Software as a service ("SaaS," typically pronounced "sass") is a software distribution model in which applications are hosted by a vendor or service provider and made available to customers over a network, typically the Internet. The SaaS vendor owns the software and runs it on computers in its data center. The customer does not own the software but effectively rents it, usually for a monthly fee. SaaS is sometimes also known as hosted software.

A Service Level Agreement ("SLA") is a negotiated agreement between two parties where one is the customer and the other is the service provider. In the situation of a SaaS arrangement this should be a legally binding formal contract. The SLA records a common understanding about services, priorities, responsibilities, guarantees, and warranties. Each area of service scope should have the "level of service" defined. The SLA may specify the levels of availability, serviceability, performance, operation, or other attributes of the service, such as billing.

2. Procedure

The purpose of this document is to provide basic guidelines of security considerations for Data Stewards when a City of Cambridge is evaluating a SaaS offering. A full risk assessment should be conducted before a final purchase decision is made. A Service Level Agreement should be executed to outline the responsibilities and risk assumptions of the service provider and those of the City of Cambridge.

Please consider the following as you review Software as a Service offerings. Use the letters in parenthesis as a guide: The higher your availability ("A") or confidentiality ("C") requirements are, the more critical the responses to these questions become.

Functionality:

_____ Does the SaaS offering of the product have all of the features of the on-site, internally installed offering of the same product?
_____ If not, are there any critical features absent from either choice?
_____ Is the vendor aware of legal discovery and retention requirements and will they comply with a litigation hold request?

Reliability:

_____ (A) What Service Level Agreement options are available from the SaaS offering?
_____ (A) Does the contract contain penalty clauses for SLA nonconformance?
_____ (A) Will the company provide metrics regarding conformance of SLAs with other

clients?
_____ (A) Do the terms of the Service Level Agreement meet your business needs?

Integration:
_____ Will the SaaS offering of the product require integration with any existing, internal, on-site applications/systems?
_____ (A) If so, what are the patching and upgrade coordination plans?
_____ (C,A) What are the network and network security requirements for such integration?

Change Management:

_____ (C,A) Are patches, service level releases and other upgrades handled consistent with expectations?
_____ (A) Are changes to the SaaS offering's environment conducted in a replica test environment before they are promoted to production?
_____ (A) Who approves such promotions? Is it approved by our organization?
_____ (A) Will we as an organization be involved in any development and testing?

Data Access:

_____ (C) How will the SaaS offering use our organization's data? Will the company use our information only as we intend them to?
_____ (C) Will we receive a copy of the SaaS offering's privacy policy? Is the privacy policy consistent with how we expect them to utilize the information?
_____ (C,A) Will the SaaS offering allow our organization to import and export data to and from the SaaS solution?
_____ (C) Will we have full access to all of our data at all times within the SaaS offering?
_____ (C) Is our data completely segregated from any other clients of the SaaS offering?
_____ (C) If our organization terminates the agreement with the SaaS offering, what happens to our data?

Data Security:

_____ (C) What are the SaaS offering's stated theft-prevention mechanisms?
_____ (C) Will our organization be notified in the event of a data breach? How will we be notified? Is the notification timely and contractually required?
_____ (C) Does the vendor conduct third party penetration and application security tests on a regular basis?
_____ (C) Will we receive third party penetration and application security test results?
_____ (C) Will the company offer legal commitments with regards to their security measures?
_____ (C) Does the SaaS offering's security controls meet all of our organization's regulatory

compliance requirements?
_____ (C) Has the vendor conducted a System and Organization Controls ("SOC") 1 Type 2[1] or other third party audit?
_____ (C) Will the vendor share the SOC 1 Type 2 results?
_____ (C) Has the vendor had any breaches within the last two years?
_____ (C) Does the vendor host any data outside of the USA? Do the countries where the data is hosted provide sufficient legal protections to ensure the confidentiality of our information?

Human Resources:

_____ (C) Are all employees of the SaaS offering's company required to sign non-disclosure and confidentiality agreements?
_____ (C) Are the employee screening policies and procedures satisfactory? (Do they conduct background or credit checks?)
_____ (C) Are employee accounts reviewed periodically for appropriate access?
_____ (C) If the SaaS offering's company outsources any job functions, what are the non-disclosure and confidentiality agreements, and employee screening requirements of the out-sourced agencies?
_____ (C) Does the vendor outsource any job functions outside of the USA? Do the countries where job functions are outsourced provide sufficient legal protections to ensure the confidentiality of our information?

Physical Security:

_____ (C) Are the SaaS offering's data center(s) access controlled sufficiently?
_____ (C,A) Are appropriate environmental controls in place in the SaaS offering's data center(s)?

Business Continuity and Disaster Recovery:

_____ (A) How frequently is the SaaS offering's system/application backed up?
_____ (A) How frequently is our data backed up by the SaaS offering?
_____ (C) Are those backups encrypted?
_____ (A) Are those backups tested?
_____ (C,A) Are those backups performed or transported off-site?

---

[1]The American Institute of CPAs ("AICPA") defines SOC 1 as follows: "Report on Controls at a Service Organization Relevant to User Entities' Internal Control over Financial Reporting ("ICFR"). These reports, prepared in accordance with AT-C section 320, Reporting on an Examination of Controls at a Service Organization Relevant to User Entities' Internal Control Over Financial Reporting, are specifically intended to meet the needs of entities that use service organizations (user entities) and the CPAs that audit the user entities' financial statements (user auditors), in evaluating the effect of the controls at the service organization on the user entities' financial statements. There are two types of reports for these engagements: Type 2 - report on the fairness of the presentation of management's description of the service organization's system and the suitability of the design and operating effectiveness of the controls to achieve the related control objectives included in the description throughout a specified period. Type 1 – report on the fairness of the presentation of management's description of the service organization's system and the suitability of the design of the controls to achieve the related control objectives included in the description as of a specified date." See: https://www.aicpa.org/interestareas/frc/assuranceadvisoryservices/aicpasoc1report.html.

EXHIBIT A: VENDOR/CLOUD/HOSTED SOFTWARE AS A SERVICE ASSESSMENT

_____ (A) Will our organization have the ability to perform our own backups of our data from the SaaS offering?
_____ (A) How quickly can the SaaS offering recover from a catastrophic failure?
_____ (A) How often are the business continuity and disaster recovery plans tested and reviewed?

EXHIBIT B: WEB SECURITY STANDARDS AND PRACTICES

City of Cambridge's
Web Security Standards and Practices

Objective and Scope

This Web Security Standards document establishes a baseline of security related requirements for all City of Cambridge-supported web services and websites, including City of Cambridge-branded applications supported/hosted by 3rd parties.

This document is intended for personnel responsible for developing, implementing and supporting City of Cambridge's web services and websites. The purpose of this document is to provide coding standards, which are based on accepted industry practices, to minimize security exploits due to improper and nonstandard coding practices. It also provides references to information about common web security vulnerabilities to enhance understanding of the root causes of such issues and how to remediate them appropriately.

1. Introduction

The materials presented in this document are obtained from the Open Web Application Security Project ("OWASP"), the SysAdmin, Audit, Network, Security Institute ("SANS"), and other recognized sources of industry best practices.

OWASP is an open community dedicated to enabling organizations to develop, purchase, and maintain applications that can be trusted. All of the OWASP tools, documents, forums, and chapters are free and open to anyone interested in improving application security.

SANS Institute was established as a cooperative research and education organization. At the heart of SANS are the many security practitioners in varied global organizations from corporations to universities working together to help the entire information security community.

This document is divided into seven sections that cover the following topics:

EXHIBIT B: WEB SECURITY STANDARDS AND PRACTICES

Section	Description
Threat Risk Modeling	Brief description of approved threat risk modeling methodologies to provide context for the application of web security standards described in the next section.
Web Security Standards	Specifies coding standards and basic security practices that must be followed when developing and improving websites and web applications.
OWASP Application Security Checklist	A checklist of key items to review and verify effectiveness.
OWASP Top 10 Application Security Risks	Issues commonly identified as susceptible to exploitation using well-known techniques, and recommended remediation approaches.
SANS Top 25 Most Dangerous Software Errors	Commonly exploited coding mistakes and recommended remediation approaches.
Additional Security Best Practices	Supplemental security controls that may optionally be considered.
References	Hyperlinks to materials referenced within this document and suggestions for further reading.

You must read all sections and implement controls which are aligned with business and operational requirements.

Before considering the specific security features and controls described in this document, it is important to understand the context for the application of web security standards. Security features and controls should be implemented to remediate meaningful risks to a web application.

City of Cambridge recommended threat risk modeling methodology is the OWASP Threat Risk Model process set forth below[2]:

Application Threat Modeling

Threat modeling works to identify, communicate, and understand threats and mitigations within the context of protecting something of value.

Threat modeling can be applied to a wide range of things, including software, applications, systems, networks, distributed systems, things in the Internet of things, business processes, etc. There are very few technical products which cannot be threat modeled; more or less rewarding, depending on how much it communicates, or interacts,

[2] The OWASP process outlined in this Policy was retrieved from: http://www.owasp.org/index.php/Threat_Risk_Modeling.

with the world. Threat modeling can be done at any stage of development, preferably early - so that the findings can inform the design.

What

Most of the time, a threat model includes:

- A description / design / model of what you're worried about
- A list of assumptions that can be checked or challenged in the future as the threat landscape changes
- A list of potential threats to the system
- A list of actions to be taken for each threat
- A way of validating the model and threats, and verification of success of actions taken

Our motto is: Threat modeling: the sooner the better, but never too late.

Why

The inclusion of threat modeling in the SDLC can help

- Build a secure design
- Efficient investment of resources; appropriately prioritize security, development, and other tasks
- Bring Security and Development together to collaborate on a shared understanding, informing development of the system
- Identify threats and compliance requirements, and evaluate their risk
- Define and build required controls.
- Balance risks, controls, and usability
- Identify where building a control is unnecessary, based on acceptable risk
- Document threats and mitigation
- Ensure business requirements (or goals) are adequately protected in the face of a malicious actor, accidents, or other causes of impact
- Identification of security test cases / security test scenarios to test the security requirements

Four Questions

Most threat model methodologies answer one or more of the following questions in the technical steps which they follow:

1. **What are we building?**

As a starting point you need to define the scope of the Threat Model. To do that you need to understand the application you are building, examples of helpful techniques are:

- Architecture diagrams
- Dataflow transitions

- Data classifications
- You will also need to gather people from different roles with sufficient technical and risk awareness to agree on the framework to be used during the Threat modeling exercise.

2. What can go wrong?

This is a "research" activity in which you want to find the main threats that apply to your application. There are many ways to approach the question, including brainstorming or using a structure to help think it through. Structures that can help include STRIDE, Kill Chains, CAPEC and others.

3. What are we going to do about that?

In this phase you turn your findings into specific actions. See OWASP Threat Modeling Outputs below.

OWASP Threat Modeling Outputs

Summary

This page details the discussions and responses prompted by the question "Are all outputs of threat modeling bugs?" as part of the OWASP Slack Threat Modeling Project.

Not all outputs are bugs, and it is difficult to classify a design problem in code (Versus an implementation problem) as a bug. An issue discovered at different times will take different forms. A trivial example would be "Spoofing" at the design stage of a web application.

When creating a user story, the user in question may think "I want secure authentication so that only I can access my data". During the development stage, "Spoofing" may appear while threat modeling - and thus considered a bug, if it appeared upon realizing that changing their account password does not require the user to input the current password.

Considerations

- Changed whiteboard diagrams
- Bugs
- Requirements (Technical or Procedural)
- New user stories in the backlog
- Using wikis at the intermediate stage
- Findings
- Classifying bugs as threats
- Organizational discipline:
    - Agreement
    - Driving discussion using threat modeling
    - Tagging

Possible Responses

1. Take the bug out of threat modeling and consider what needs to happen next, weighting the risk vs value for the bug and the attention it receives as part of prioritization.
2. Allow whoever creates priorities during threat modeling to contextualize the risk, then make the decision on what should be immediately prioritized, what should be put aside, and what research is necessary for further focus later on.

Response 2 is preferable to response 1.

Follow Up

Deliver the decisions and changes to:

- Software
- Operations/IT
- Risk management

4. Did we do a good enough job?

Finally, carry out a retrospective activity over the work you have done to check quality, feasibility, progress, and/or planning.

Process

The technical steps in threat modeling involve answering questions:

- What are we working on? What can go wrong? What will we do with the findings?
- Did we do a good job? The work to answer these questions is embedded in some sort of process, ranging from incredibly informal Kanban with Post-its on the wall to strictly structured waterfalls.

The effort, work, and timeframes spent on threat modeling relate to the process in which engineering is happening and products/services are delivered. The idea that threat modeling is waterfall or 'heavyweight' is based on threat modeling approaches from the early 2000s. Modern threat modeling building blocks fit well into agile and are in wide use.

When to Threat Model

When the system changes, you need to consider the security impact of those changes. Sometimes those impacts are not obvious.

Threat modeling integrates into Agile by asking "what are we working on, now, in this sprint/spike/feature?"; trying to answer this can be an important aspect of managing security debt, but trying to address it per-sprint is overwhelming. When the answer is that the system's architecture isn't changing, no new processes or dataflows are being introduced, and there are no changes to the data structures being transmitted, then it is unlikely that the answers to 'what can go wrong' will change. When one or more of those changes, then it's useful to examine what can go wrong as part of the current work package, and to understand designs trade-offs you can make, and to understand what you're going to address in this sprint and in the next one. The question of did we do a good job is split: the "did we address these threats" is part of sprint delivery or merging, while the broader question is an occasional saw-sharpening task.

After a security incident, going back and checking the threat models can be an important process.

Threat Modeling: Engagement Versus Review

Threat modeling at a whiteboard can be a fluid exchange of ideas between diverse participants. Using the whiteboard to construct a model that participants can rapidly change based on identified threats is a high-return activity. The models created there (or elsewhere) can be meticulously transferred to a high-quality archival representation designed for review and presentation. Those models are useful for documenting what's been decided and sharing those decisions widely within an organization. These two activities are both threat modeling, yet quite different.

After you've performed the risk evaluation, you need to consider the controls to implement. To determine the type of security control that is needed, you should apply security control requirements using the Confidentiality, Integrity, Availability, and Accountability (CIAA) methodology as follows:

1) Determine whether a security control mechanism is required to ensure the Confidentiality, Integrity, Availability and/or Accountability of the data.

2) Using the CIAA approach, evaluate and rank the importance of each to prioritize what and where control mechanisms should be applied.

3. Web Security Standards

This section lists the web security standards which must be implemented by City of Cambridge supported web applications, services, and sites.

3.1 Deny access for exception conditions

Handling errors securely is critical in secure coding; especially exceptions that occur in the processing of a security control. It is important that these exceptions do not enable behavior that the established control would normally not allow. There are only three possible outcomes from a security mechanism:

1) allow the operation
2) disallow the operation
3) exception

In general, you must design your security mechanism so that a failure will follow the same execution path as disallowing the operation. For example, security methods like is Authorized (), is Authenticated (), and validate () should all return false if there is an exception during processing.

3.2 Validate all inputs and sanitize all outputs

All input data must be validated, input data validation should occur in the following sequence:

1) Decode the data before performing the validation – for example, check input strings to prevent the program from executing malicious commands, scripts, codes, etc.;

2) Check for length criteria - for example, determine if it is within the allowable predetermined minimum and maximum range;

3) Check for acceptable data types - for example, determine if it is a valid data type (e.g., characters or numbers only); and finally

4) Check for unacceptable data types – for example, determined whether data entered is non-characters, non-numeric, special characters.

All outputs must be sanitized to ensure outputs do not reveal too much information, especially for error messages which can provide too much information (e.g., default system generated messages) that an attacker can use to exploit security weaknesses. For error handling of input data, the error message should not reveal too much information.

For example, when there is an invalid user ID and/or password entered, the error message should not reveal what component entered, whether it's the user ID or password, which caused the error. The message should be general (e.g., invalid entry) and not reveal more information than necessary.

3.2 Maintain separation of duties

The concept of separation of duties is that the entity that approves an action, the entity that carries out an action, and the entity that monitors that action must be distinct. The goal is to eliminate the possibility of a single user from carrying out and concealing a prohibited action.

In general, application administrators should not be users of the application because application administrators inherit privileged access in the application. There are situations where the application administrator is also an application user. In such scenario, maintain

separate accounts, one as the application administrator and one as the application user. The use of the application administrator account must be used exclusively for authorized administrative tasks only.

3.3 Verify all user authentication and authorization

There must be a security control mechanism to authenticate the identity of the user. This is typically handled with a User ID account and a password. The password must be sufficiently long and complex, consisting of alphanumeric characters and, if feasible, special characters. Authentication can be achieved using City of Cambridge Active Directory Services.

After the user is authenticated, a security control mechanism must also ensure that the user's access rights to the data must be limited to only his authorized access level. Implement user access with the least privilege required.

3.5 Assign user with the least privilege access level

Access to resources must be granted with the least privilege to ensure that accounts have the least amount of privilege required to perform their job function and responsibility. This encompasses user rights and resource permissions, including file and database permission. The user's access rights and privileges must be limited to perform only tasks that the user is authorized and not beyond his authority. Also, before the access is granted, ensure that appropriate authorization is obtained from the requestor's

manager and the data owner or data steward (i.e., a person who has been given the authority by the data owner to approve data access on behalf of the owner).

For example, if the user only requires read access to the application, then this is all the permission that should be granted. Under no circumstances should the user be allowed update privileges unless he has been explicitly authorized for both read and update access.

3.6 Establish secure default settings

Security related parameters settings, including passwords, must be secured and not user changeable. For example, by default, password length and complexity should be enabled and not be changed by the user. Also, some applications are supplied with one or more default user accounts and passwords. If these accounts are not used, then they should be disabled or removed. If the default accounts are needed, then the default passwords must be changed immediately to a new complex password.

If your application requires a default password either for the user to initially sign-on to the application or in the event of a password reset for forgotten password, then the default password must be a complex password and should be different for each user. The default

password should also have an expiration period (e.g., not valid more than 24 hours) and onetime use only; thus, the system will force the user to change his password immediately after using the default password.

3.7 Keep the attack surface area to a minimum

The system attack surface is the collection of input points that the application has for an attacker. More points of entry into the application provide more avenues for an attacker to find a weakness. Every feature that is added to an application adds a certain amount of risk to the overall application. The aim for secure development is to reduce the overall risk by reducing the attack surface area. Each feature must function accordingly to application specification and business requirements and should not be allowed to perform functions beyond its intended purpose.

3.8 Keep security simple and avoid security by obscurity

Attack surface area and simplicity go hand in hand. Some programmers prefer overly complex approaches to what would otherwise be relatively straightforward and simple code. Do not add complexity when simplicity will achieve the same results. You must avoid the use of complex architectures when a simpler approach would be faster and more efficient.

Using obscurity to provide security control always fails when it is the only control. This is not to say that keeping secrets is a bad idea. It simply means that the security of key systems should not be reliant upon keeping details hidden. In other words, security plus obscurity is fine; obscurity by itself is not.

For example, the security of an application should not rely upon knowledge of the source code being kept secret. The security should rely upon many other factors, including reasonable password policies and controls, defense in depth, transaction limits, network access controls, and audit trails.

3.9 Provide defense in depth

Defense in depth is a concept where one control would be reasonable, more controls that approach risks in different fashions are better. Controls, when used in depth, can make severe vulnerabilities very difficult to exploit and thus unlikely to occur. The more control layers, the more difficult it would be to circumvent them than a single control; but you should also be

mindful that security controls should not be too complex making them difficult to manage and maintain.

With secure coding, this may take the form of combining controls from various elements such as: tier-based validation, centralized auditing controls, and requiring user activity to be logged.

3.10 Do not automatically trust access from other systems

Implicit trust of externally run systems is not warranted; therefore, system access validation must be checked when a request is initiated by the external system. All external systems should be treated in a similar fashion.

For example, when receiving requests from another system, the identity of the system must be validated and the permission checked before processing the requests. If inputs are received from the system, always validate the input before processing.

3.11 Maintain up-to-date security fixes and patches

Once security issues are identified, vendors will release security fixes or patches to prevent exploits of the vulnerability. Therefore, you must constantly review the security notifications and updates from the vendor and apply the security fixes and patches on a timely basis.

3.12 Maintain Audit Logs

You will never be able to prevent every attack. No matter how good your defenses, some things will slip through. It is critical that there be sufficient audit logs in place to be able to discover that an attack occurred.

Audit Logging procedures must be documented. Audit logs recording user activities, exceptions, and information security events must be produced.

Where technically feasible, recorded events must include: Security administration activities; Restricted account access; Logon/logoff success and failure; Unsuccessful attempts to access systems or information; Dates and times of activity; Source of connection and access.

Logs are a target and therefore, they must be secured. Wherever possible, lock down log files so that only administrators have access to them. Ideally, spool logs to a separate log server. Logs must be protected to prevent the following: alterations to the recorded messages; editing or deletion of log files; log storage capacity being exceeded; viewing of confidential information in logs by unauthorized individuals.

4. OWASP Web Application Security Checklist

The “OWASP Application Security Verification Standards 2014 - Web Application Standards” provides a security checklist of keys controls for you to verify.

These are a few examples of control points to check:

- Verify that all pages and resources require authentication except those specifically intended to be public.
- Verify that all password fields do not echo the user’s password when it is entered, and that password fields (or the forms that contain them) have autocomplete disabled.
- Verify that sessions are invalidated when the user logs out.
- Verify that users can only access URLs for which they possess specific authorization.

At a minimum, you must perform level 1 verifications of OWASP Application Security Verification Standards 2014 to ensure appropriate security measures are implemented.

5. OWASP Top 10 Application Security Risks

The "OWASP Top 10 - 2010 The Ten Most Critical Web Application Security Risks" document which OWASP has identified the ten most critical web application security risks and suggested remediation to implement.

You must read and understand the "OWASP Top 10 - 2010 The Ten Most Critical Web Application Security Risks" set forth below.

OWASP TOP 10

The OWASP Top 10 is a standard awareness document for developers and web application security. It represents a broad consensus about the most critical security risks to web applications.

Globally recognized by developers as the first step towards more secure coding. Companies should adopt this document and start the process of ensuring that their web applications minimize these risks. Using the OWASP Top 10 is perhaps the most effective first step towards changing the software development culture within your organization into one that produces more secure code.

Top 10 Web Application Security Risks

- **A1:2017-Injection**: Injection flaws, such as SQL, NoSQL, OS, and LDAP injection, occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization.

- **A2:2017-Broken Authentication**: Application functions related to authentication and session management are often implemented incorrectly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users' identities temporarily or permanently.

- **A3:2017-Sensitive Data Exposure**: Many web applications and APIs do not properly protect sensitive data, such as financial, healthcare, and PII. Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes. Sensitive data may be compromised without extra protection, such as encryption at rest or in transit, and requires special precautions when exchanged with the browser.

- **A4:2017-XML External Entities (XXE)**: Many older or poorly configured XML processors evaluate external entity references within XML documents. External entities can be used to disclose internal files using the file URI handler, internal file shares, internal port scanning, remote code execution, and denial of service attacks.

- **A5:2017-Broken Access Control**: Restrictions on what authenticated users are allowed to do are often not properly enforced. Attackers can exploit these flaws to access unauthorized functionality and/or data, such as access other users' accounts, view sensitive files, modify other users' data, change access rights, etc.

- **A6:2017-Security Misconfiguration**: Security misconfiguration is the most commonly seen issue. This is commonly a result of insecure default configurations, incomplete or ad hoc configurations, open cloud storage, misconfigured HTTP headers, and verbose error messages containing sensitive information. Not only must all operating systems, frameworks, libraries, and applications be securely configured, but they must be patched/upgraded in a timely fashion.

- **A7:2017-Cross-Site Scripting XSS**: XSS flaws occur whenever an application includes untrusted data in a new web page without proper validation or escaping, or updates an existing web page with user-supplied data using a browser API that can create HTML or JavaScript. XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.

- **A8:2017-Insecure Deserialization**: Insecure deserialization often leads to remote code execution. Even if deserialization flaws do not result in remote code execution, they can be used to perform attacks, including replay attacks, injection attacks, and privilege escalation attacks.

- **A9:2017-Using Components with Known Vulnerabilities**: Components, such as libraries, frameworks, and other software modules, run with the same privileges as the application. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications and APIs using components with known vulnerabilities may undermine application defenses and enable various attacks and impacts.

- **A10:2017-Insufficient Logging & Monitoring**: Insufficient logging and monitoring, coupled with missing or ineffective integration with incident response, allows attackers to further attack systems, maintain persistence, pivot to more systems, and tamper, extract, or destroy data. Most breach studies show time to detect a breach is over 200 days, typically detected by external parties rather than internal processes or monitoring.

These are the areas where majority of security exploits has been identified that can occur and therefore, preventive measures must be implemented.

You must implement the OWASP recommended remediation action items which counteract the exploits and vulnerabilities.

6. SANS Top 25 Most Dangerous Software Errors

The SANS Institute's ("SANS") Top 25 Most Dangerous Software Errors list (the "SANS Top 25") is another list that identifies the most prevalent software errors that can result in exploits and vulnerabilities.

EXHIBIT B: WEB SECURITY STANDARDS AND PRACTICES

The SANS Top 25 are as follows[3]:

1. Improper Restriction of Operations within the Bounds of a Memory Buffer
2. Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')
3. Improper Input Validation
4. Information Exposure
5. Out-of-bounds Read
6. Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')
7. Use After Free
8. Integer Overflow or Wraparound
9. Cross-Site Request Forgery (CSRF)
10. Improper Limitation of a Pathname to a Restricted Directory ("Path Traversal")
11. Improper Neutralization of Special Elements used in an OS Command ("OS Command Injection")
12. Out-of-bounds Write
13. Improper Authentication
14. NULL Pointer Dereference
15. Incorrect Permission Assignment for Critical Resource
16. Unrestricted Upload of File with Dangerous Type
17. Improper Restriction of XML External Entity Reference
18. Improper Control of Generation of Code ("Code Injection")
19. Use of Hard-coded Credentials
20. Uncontrolled Resource Consumption
21. Missing Release of Resource after Effective Lifetime
22. Untrusted Search Path
23. Deserialization of Untrusted Data
24. Improper Privilege Management
25. Improper Certificate Validation

You must review and implement the SANS recommended remediation action items which counteract the exploits and vulnerabilities.

7. Additional Security Best Practices

The following are additional security best practices that you should consider implementing, if feasible.

- Provide a mechanism to ensure timeliness of data access. After the validation for data access is performed, it is expected that the data would be used immediately. However, if there is a time lapse between data access validation and data use, then a re-validation must be performed before granting access. For example, there should be a mechanism to monitor user inactivity time and require the user to revalidate after exceeding the allowable threshold. This can assure that the person has not walked away and an unauthorized person is using his account.

[3] The SANS Top 25 outlined in this Policy were retrieved from: http://www.sans.org/top25-software-errors/

EXHIBIT B: WEB SECURITY STANDARDS AND PRACTICES

- Identify what permission level is actually required for database table access by the application and limit access accordingly. For example, if the application only requires read access, do not allow it update access. Implement least privilege required even for the application.

- DBA should access the database the same way as the application to ensure activity is logged and audited. DBA should access database via stored procedures to track their activity.

- Separate the web server from the application server (i.e., the web server should be physically or logically/virtually separate from the application server).

- Separate application from the database (i.e., the application server should be physically or logically/virtually separate from the database server).

- When performing a security evaluation process, involve all parties from technology, operational, and business areas with vested interest. Perform security process assessment by examining each component in detail:

  1) Input control;
  2) Output control;
  3) Authentication control;
  4) Authorization control;
  5) Session management control;
  6) Logging and auditing;
  7) Use of encryption.

- Keep security simple and configurable:

  - Plan security configuration from the beginning;
  - Separate users and administrator logins;
  - Practice least privilege – even in development;
  - Configuration must be simple;
  - Keep things secure by default;
  - Configure logging levels;
  - Clearly document all security configurations.

- Document program source codes and ensure that documentation is maintained and kept up to date.

- Review log activity and audit the logs for exceptions.