NuHarbor SECURITY

VERMONT Secretary of State

Vermont Secretary of State

Elections Platform

Penetration Test Report

06/22/2016

Version Control

APPLICATION PENETRATION TESTING	
Client Name	*Vermont Secretary of State*
Client Contacts	*Steve Mattera*
Document Issue No	*1.0*
Authors	*Hunter Gregal*
Approved by	*Nate Couture*
Delivery Date	*06/22/2016*
Data Classification	*Client Confidential*

Table of Contents

Executive Summary

NuHarbor Security Overview

NuHarbor Security (NuHarbor) is a national leader of security, risk, and compliance management solutions and services. NuHarbor was founded on the belief that Information Security and Risk Management can be positioned as a business enabler, thus creating a competitive advantage. We promote a new understanding of Information Security Governance through intelligently applying business controls that reduce business risk, reduce compliance overhead to client staff, and effectively deploy security technology that reduces technology debt. As an "Enterprise Scale" Information Security Firm, NuHarbor provides end-to-end services for all types and sizes of organization and industry, including Fortune 500 companies and Federal entities.

Objective

NuHarbor's overall objective for this engagement is to ensure that the Vermont Secretary of State Elections Platform web application is designed and protected with appropriate information security controls to ensure the integrity, confidentiality, and availability of Vermont Secretary of State information. For the purposes of this engagement, NuHarbor used the Vermont Secretary of State's User Acceptance Testing (UAT) environment. In order to conduct this evaluation NuHarbor leveraged various tools, techniques, and manual attacks:

1. A full crawl of the web application to determine what pages were available and identify hidden pages.
2. Use of [redacted] and [redacted] software to perform browser and server data transmission analysis, manual Cross-Site Scripting (XSS) and Cross-Site Request Forgery (CSRF) attacks, injection attacks, session management attacks, and business logic abuse attacks.
3. Use of [redacted] dynamic web application scanner to identify Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF), injection, and security misconfiguration vulnerabilities in an automated fashion.
4. Use of [redacted] to automate the validation of identified SQL Injection vectors
5. Use of additional targeted tools to further explore and attempt to exploit specific potential vulnerabilities.

The work performed provides a practical approximation of the weak points in the Elections Platform web application and infrastructure, describes the risks of such vulnerabilities, and facilitates the design of a plan for implementing countermeasures based on priorities established by Vermont Secretary of State. These priorities are determined according to Vermont Secretary of State's security goals and the quantity and severity of the vulnerabilities found. It is important to note, however, that this project should not be considered a full audit of Vermont Secretary of State's security posture, nor should it be thought of as an in-depth analysis of the possibilities to breach it.

Security Testing Report

Testing Approach

NuHarbor understands that Vermont Secretary of State's overall objective is to ensure that appropriate information security controls are implemented within its major environments, applications and computing platforms to preserve integrity, confidentiality, and availability of its information and computing resources. Effective implementation of these security controls will aid in the prevention of unauthorized, accidental, or deliberate disruption, disclosure, modification, and use of Vermont Secretary of State' information technology resources.

The penetration testing effort was focused on but not limited to:

- Evaluation of the web application against the OWASP Top 10 Vulnerabilities
- Cross role and cross customer data access and manipulation
- Cross user attack vectors
- Arbitrary remote code execution
- Application-specific logic manipulation

NuHarbor conducted Web Application Penetration Testing against the following sites:

- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]

The testing was completed with the following application user roles:

- [redacted]
  - [redacted]
  - [redacted]
- [redacted]
  - [redacted]
  - [redacted]
  - [redacted]
  - [redacted]
- [redacted]
  - [redacted]
  - [redacted]
  - [redacted]
- [redacted]
  - [redacted]
- [redacted]
  - [redacted]

The assessment focused on advanced manual exploit methodologies and techniques. To support the manual testing activities [redacted] and [redacted] were utilized as a method to intercept and manipulate traffic between the browser and web server. A combination of [redacted] to evaluate potential vulnerabilities. Additional testing

activities included [redacted] to identify specific vulnerabilities or configuration issues.

Testing Overview

Tests looked for the following issues, among others:

- **Review of session management**: focused on verifying that proper tracking of the user is performed throughout the application.
- **Authentication/authorization and communication mechanisms**: aimed at examining that proper authentication is in place and that authorization controls are applied to application user's actions.
- **Information leakage**: intended at determining if confidential information or information that might otherwise aid an attacker is disclosed by the application or its environment.
- **Input validation**: verifies that all user input is correctly validated, and sanitized if necessary, to ensure that the application behaves as expected independently of the submitted input.
- **Output encoding mechanisms**: must be correctly enforced by the application to ensure a consistent interpretation of the application's output.
- **Filtering layers**: focused on verifying that the necessary filtering mechanisms are in place to proactively defend against common web application attacks.
- **Parameter passing**: testing that all parameter handling is performed in a secure manner. For example, looking for authorization information mishandled by the application, which instead of being stored server-side is sent by the user.
- **Application logic flow**: aimed at verifying that the intended application flow is enforced by the application (i.e. that an attacker is not able to control the application flow at will, for example, bypass controls).
- **Cross-site scripting**: aimed at identifying cross-site scripting vulnerabilities throughout the application due to improper encoding of user supplied input.
- **SQL injections**: focused on determining when user input is used to construct database queries and testing the possibility of specially crafting input to control the queries, beyond the programmer's intention.
- **Path traversals**: aimed at identifying when user input is used to construct file paths and attempting to specially craft user input to escape the directory structure imposed by the application.
- **XML and Xpath injections**: determining user input used to construct XML or Xpath queries and verifying if it is possible to inject XMLtags or modify the Xpath query.
- **Integer underflow/overflow problems**: aimed at identifying such conditions when dealing with numeric user input.
- **Buffer overflow causing conditions**: verifying that proper bounds checking are performed when handling data.

Findings Overview

Positive Observations

- Technologies/components in place with no known inherent version vulnerabilities
- Generally strong authentication and session management implemented on page views
- Anti-Cross-Site Request Forgery cookie is present which suggest that attempts have been made to mitigate CSRF as an attack vector

Opportunities for Improvement

- [redacted]

Conclusion

The presence of up-to-date frameworks, session management of web pages, and Anti-Cross-Site Request Forgery token cookies show that there has been obvious security consideration during the design and deployment of the Secretary of State's Election Platform application. These security measures in place decrease the potential attack vector for the application as a whole. Unfortunately, the lack of [redacted]

Implementing stronger sanitization methods on both application reflected values and database queries would remedy a large portion of the findings discovered during testing. In addition to proper sanitization, [redacted]

The greatest security risk posed to the application results from [redacted] It is highly recommended that these vulnerabilities are remediated prior to moving the application to a production state.

Appendix A – Detailed Findings

Risk Rating	Description	Count Observed
H - High	Mission critical for Vermont Secretary of State, exploitation causes serious impact to the organization.	21
M - Moderate	Rated business critical for the Vermont Secretary of State, exploitation may cause impact to the organization.	25
L - Low	Small chance of exploitation / little risk if exploited.	11

<This is space intentionally left blank>