Voter Registration Application Review Report

Dec 16, 2019 -- Version 1.2

Proprietary Notice

Executive Summary

Synopsis

The Professionals Guild engaged Enyk Limited to perform an assessment on Voter Registration web application. The engagement was conducted as a source code review with a focus on privacy and data confidentiality.

Application Design

The primary objective of the Voter Registration application is to guide users to fill the forms for the Registration and Electoral Office (REO). With the application, users can fill in their personal information and e-signature. The application will then generate a PNG image of a properly filled, signed form. The image can be saved on the user's own computer and the user can then submit it to REO.

The application is designed to work purely on browser-side. Thus no personal data will be sent to or stored by any server. The personal data can only be stored in users' computers when they requested to download the completed form.

Scope

The testing team reviewed the target repository[1] with commit `e144b31` made on July 26, 2019. The assessment assumes malicious attackers will try to retrieve users' personal data entered in the application. As a client-side application, all remote servers, including the host server, are untrusted and no remote computer should process any data. The team investigated if any filled information were sent to an external party.

Limitations

As the assessment is in the form of white-box source code review, the team did not validate any potential vulnerabilities and issues induced by the hosting infrastructure. If the hosting infrastructure was compromised, the application code can be altered by an attacker to steal sensitive user data, even the application has been audited and considered safe.

The Voter Registration application included third-party libraries jQuery 1.11.3 and Bootstrap 3.3.5 Any undiscovered vulnerabilities of the aforementioned packages were not included in scope.

[1] https://github.com/yulapshun/voter-registration

Methodology

The testing team deployed the following test plan for the code review. Primarily, the team focuses on the Javascript used for the web application, including both the inline or imported scripts. From the source code review, the team determines if any filled information is sent to a first- or third-party including web analytics services. For the imported libraries such as jQuery, the team checks if there are existing vulnerabilities for the version and investigate if they are applicable to the application. Moreover, the team checks for any vulnerabilities induced by controlling the parameters of query strings from URLs.

Key Findings

A code review exercise is conducted on the Voter Registration web application codebase, and no significant risk items were found in this review.

Two outdated code libraries are used by the project. They are recommended to be upgraded to avoid the possible vulnerabilities.

Security Recommendations

The team recommends upgrading the obsolete dependencies as soon as possible.

The team recommends hosting the application on a trustworthy file hosting service, such as Google Cloud Storage, Amazon Web Service S3, or Github Pages. The web hosting service shall use HTTPS to protect the integrity of the application. It is to prevent, for instance, an adversary from injecting malicious codes to the webpage. The team further suggests that the hosting infrastructure should undergo some form of security review before the application is ready to release.

The team also recommends regularly search for any phishing sites similar to the application and report to blacklist services such as Google Safe Browsing and PhishTank.

Updates as of Dec 16, 2019

The Professionals Guild concluded not to fix the low-risk issues in this report because they do not affect the current version of the software. The Enyk team agrees with this conclusion. Our team further confirms on Dec 16, 2019 that other issues in this report are resolved to our satisfaction.

Vulnerability Details

Table of Vulnerability

Findings	Number of findings
Critical risk issues	0
High risk issues	0
Medium risk issues	0
Low risk issues	2
Informational issues	0
Functional issues	2
Total	**4**[2]

Severity	Issue	Tally	Risk score[#]
Low	Obsolete jQuery version	1	1*3 = 3
	Obsolete bootstrap version	1	1*3 = 3
Functional	Uncheck “Provide email to candidates” does nothing	1	0
	Use of Google Analytics not cleaned up	1	0
Total		**4**	**-**

*# Risk score = Probability * Impact. Risk score table available on Appendix A*

[2] The Enyk team confirms on Dec 16, 2019 that issues in this report are resolved to our satisfaction.

Evaluation of Security Engineering Goals

Privacy protection

Evaluation:
Voter Registration does an excellent job to protect the end-users' privacy by:

- Minimizing server-client interaction
  There are zero interactions between the client and the server, apart from requesting the resources such as HTML, Javascript, CSS and fonts. This has largely reduced the attack surfaces.
- Minimizing data storage
  The application does not save any personal data locally nor remotely except users have requested to download the completed form to the device. This reduces the potential attack target and ensures that the sensitive personal information (SPI) will not be stored anywhere except on the trusted devices.

Recommendation:

- Continue to avoid unnecessary data storage and interaction in further development.

Mitigate reflected or stored attack vectors

Evaluation:
Voter Registration does an excellent job to avoid malicious states pollutions:

- Minimized the usage for the URL parameters.
  - For `app.html`, the only parameter that is used is `type`. `type` is parsed by `getApplicationType` in `app.js` and needs to be one of the four possible values: `new-district`, `new-functional`, `change-address` or `change-functional`. Otherwise, it will simply fallback to the default: `new-district`. Hence it is impossible to inject malicious `type`s.
  - For `index.html`, `note.html` and `terms.html`, no URL parameters are used.
- Designed as a client-side only application. Without the use of `cookie` and `localStorage`, eliminates the means to perform stored XSS attacks.

Secure coding practices

Evaluation:
The developers have been aware of possible XSS issues. Building a client-side only application mitigates stored-XSS and reflected-XSS. Moreover, the developers have used `.text()` in place of `.html()` for user inputs which mitigated the risk of evaluating unsafe scripts.

Source code transparency

Evaluation:
The first-party HTML and Javascript files of the application are not minified. This is a good practice that enables end-users to review the source code and verifies the security of the

application. Allowing reviews by the public help improve the credibility and also the security of the application.

Low Risk Issues

Vulnerability: Obsolete jQuery version

Risk:
Low (Impact: Medium, Likelihood: Rare)

Target:
`jquery.min.js`

Impact:
The use of obsolete library version can induce security vulnerabilities.

Description:
jQuery v1.11.3, which is released 4 years ago, is used in the application. In the corresponding version, a cross-site scripting (XSS) and a prototype pollution vulnerability were reported. While the application is not vulnerable to those issues, it is recommended to upgrade the jQuery version to reduce potential risks.

The Common Vulnerabilities and Exposures (CVE) repository lists the vulnerabilities reported against jQuery[3] [4] having CVSS scores of 5.4 and 5.6 respectively.

Recommendation:
The team recommends following a continuous integration process with a package manager, such as `npm` or `yarn`, to keep dependencies updated.

Update jQuery from v1.11.3 to the latest release, v3.4.1 at the time of writing. As there are breaking changes from major versions, we suggest checking the jQuery upgrade guide[5] or using the jQuery Migrate plugin[6] for easier migration, to comply with the latest coding standards.

Updates as of Dec 16, 2019:
The Professionals Guild concluded not to fix this issue because it does not affect the current version of the software. The Enyk team agrees with this conclusion.

---

[3] https://www.cvedetails.com/vulnerability-list/vendor_id-6538/product_id-11031/version_id-286367/Jquery-Jquery-1.11.3.html

[4] https://snyk.io/test/npm/jquery/1.11.3

[5] https://jquery.com/upgrade-guide/3.0/

[6] https://github.com/jquery/jquery-migrate

Vulnerability: Obsolete bootstrap version

Risk:
Low (Impact: Medium, Likelihood: Rare)

Target:
`bootstrap/js/bootstrap.min.js`

Impact:
The use of obsolete library version can induce security vulnerabilities.

Description:
Bootstrap v3.3.5, which was released 4 years ago, is used by the application. In the corresponding version, various cross-site scripting (XSS) vulnerabilities were reported.

While the application is not vulnerable to those issues, it is recommended to upgrade the Bootstrap version to reduce potential risks.

The recommendation session contains detailed recommendations for updating bootstrap.

The CVE repository lists 5 vulnerabilities reported against bootstrap[7] [8] while having a CVSS score of 6.5.

Recommendation:
The team recommends following a continuous integration process with a package manager, such as `npm` or `yarn`, to keep dependencies updated.

Update Bootstrap from v3.3.5 to the latest release, v3.4.1 at the time of writing.

Updates as of Dec 16, 2019:
The Professionals Guild concluded not to fix this issue because it does not affect the current version of the software. The Enyk team agrees with this conclusion.

[7] https://www.cvedetails.com/vulnerability-list/vendor_id-19522/product_id-51406/version_id-286027/Getbootstrap-Bootstrap-3.3.5.html

[8] https://snyk.io/test/npm/bootstrap/3.3.5

Functional Issues

Unchecking “Provide email to candidates” does nothing

Risk:
None (Functional)

Target:
`app.js`

Source:

```
voterRegistration.setRadio = function(){
  if (this.id == "email-to-candidate-yes") {
    voterRegistration.data[this.name] = "✔";
    return false;
  }
  if (this.id == "email-to-candidate-no") {
    return false;
  }
}
```

`voterRegistration.data[this.name]` is not unset when `setRadio()` is called with `this.id` being `"email-to-candidate-no"`.

Impact:
In step 5 while filling REO-1 or REO-41, or step 6 while filling REO-2. The users will be asked if they agree to provide the email address to the candidates. Once the users agree, unchecking the field would not affect how the form is rendered.

Updates as of Dec 16, 2019:
The Enyk team confirms that this issue is resolved to our satisfaction.

Use of Google Analytics not cleaned up

Risk:
None (Functional)

Target:
`index.html`

Source:
```
{
  "vars": {
    "account": "UA-72771086-1"
  },
  "triggers": {
    "trackPageview": {
      "on": "visible",
      "request": "pageview"
    },
    "trackClickOnApplink" : {
      "on": "click",
      "selector": ".applink",
      "request": "pageview",
      "vars": {
        "documentLocation": "https://kbfl-campaign.github.io/app.html"
      }
    }
  }
}
```

Impact:
Some of the traffic from `index.html` may still be tracked and analysed by Google Analytics.

Updates as of Dec 16, 2019:
The Enyk team confirms that this issue is resolved to our satisfaction.

Revision History

Revision	Date	Changes
1.0	Aug 19, 2019	Initial document
1.1	Aug 20, 2019	Errata
1.2	Dec 16, 2019	Status updates

--- END OF REPORT ---

Appendix A. Risk Modeling

During the analysis of identified risk items, a consistent risk modelling approach based on qualitative analysis is used to portray the risk items based on factors such as confidentiality, integrity, and availability against likelihood and impact severity.

			Impact				
			Trivial	Minor	Moderate	Major	Extreme
			1	**2**	**3**	**4**	**5**
Likelihood	Rare	**1**	1	2	3	4	5
	Unlikely	**2**	2	4	6	8	10
	Moderate	**3**	3	6	9	12	15
	Likely	**4**	4	8	12	16	20
	Very likely	**5**	5	10	15	20	25

The score of a risk item is calculated by multiplying its likelihood score and impact severity score. A higher score reflects a higher risk.

The risk items are then classified as critical, high, medium, low or informational based on the following classification scheme:

Informational	**Low**	**Medium**	**High**	**Critical**
1 ≤ score < 3	3 ≤ score < 5	5 ≤ score < 15	15 ≤ score < 25	score = 25

The critical risk items require immediate attention and prudent rectification actions. High-risk items requires to be rectified as higher priority. The rectification actions for medium risk items can be prioritized after the high-risk items depending on the resources availability. Attention for low-risk items can be put at last. A longer duration can be allocated to rectify the low-risk items.