Mémoire sur le vote par Internet

Date

3 novembre 2019

Auteur

Olivier Arteau

Table des matières

Glossaire

Terme	Définition
Accessibilité	L'accessibilité dans ce document désigne tout ce qui rend plus facile l'utilisation d'une application web par l'ensemble de la population. Ceci inclut toutes les mesures pour faciliter son utilisation par des personnes avec des handicaps (ex.: support de navigation par clavier, support de lecteur d'écran, choix des couleurs et contrastes, etc.).
Application web	« En informatique, une application web (aussi appelée web application, de l'anglais) est une application manipulable directement en ligne grâce à un navigateur web et qui ne nécessite donc pas d'installation sur les machines clientes, contrairement aux applications mobiles. »[1]
Bibliothèque logicielle	« En informatique, une bibliothèque logicielle est une collection de routines, qui peuvent être déjà compilées et prêtes à être utilisées par des programmes. Les bibliothèques sont enregistrées dans des fichiers semblables, voire identiques aux fichiers de programmes, sous la forme d'une collection de fichiers de code objet rassemblés accompagnée d'un index permettant de retrouver facilement chaque routine. »[2]
Déni de service	« Une attaque par déni de service (abr. DoS attack pour Denial of Service attack en anglais) est une attaque informatique ayant pour but de rendre indisponible un service, d'empêcher les utilisateurs légitimes d'un service de l'utiliser. »[3]
Vulnérabilité logicielle	Une vulnérabilité logicielle est une faiblesse dans un système informatique qui affecte la confidentialité de ces données, l'intégrité de ces données ou la disponibilité de celui-ci.

[1] https://fr.wikipedia.org/wiki/Application_web
[2] https://fr.wikipedia.org/wiki/Biblioth%C3%A8que_logicielle
[3] https://fr.wikipedia.org/wiki/Attaque_par_d%C3%A9ni_de_service

Sommaire

Avant-propos

Ce mémoire a été écrit à titre personnel et n'engage en rien les employeurs des auteurs et des signataires. Ce mémoire aborde le sujet du vote par Internet sous un angle technique.

À propos de l'auteur

Olivier Arteau est un chercheur indépendant en sécurité informatique. Il est titulaire de nombreuses certifications et diplômes dans le domaine de la sécurité informatique et du développement logiciel (OSCE, OSCP et Baccalauréat en génie logiciel de l'ÉTS). Il a aussi été conférencier dans plusieurs conférences de sécurité informatique comme NorthSec, AppSecCali, Hackfest et GoSec.

Motivation

Je trouvais important comme expert en sécurité de soumettre un mémoire sur la question du vote par Internet, parce que je considère qu'il est toujours pertinent que des experts se penchent sur les questions de société dans lesquelles ils ont une expertise. Aussi comme je n'ai aucun lien avec des compagnies qui développent des systèmes de votation ou qui seraient susceptibles de gagner de futurs contrats liés à ce projet, l'opinion que je présente dans ce document est impartiale. De plus, même si c'est mon opinion qui est présentée dans ce mémoire, beaucoup de références sont présentes.

Position

La position qui est présentée dans ce mémoire est la suivante;

Je ne pense pas que le domaine du développement logiciel et le domaine de la sécurité informatique sont actuellement assez matures pour que l'on soit capable de développer et opérer un système de votation par Internet qui respecte les attentes que l'on devrait avoir par rapport à ce type de système. Les enjeux et problématiques que je souhaite mettre en valeur sont regroupés en quatre grandes catégories ("Difficultés actuelles à développer du logiciel", "Installations domestiques non adaptées et/ou difficiles à supporter", "Angle de compromission très large" et "Transparence difficile à obtenir pour les citoyens") qui sont détaillées dans ce mémoire.

Je recommande de ne pas aller de l'avant avec le projet de vote par Internet à court terme et de réévaluer les enjeux mentionnés dans ce mémoire à chaque 4 ans. Plusieurs des enjeux sont difficiles à adresser et dans certains cas il n'existe pas de solution théorique ou pratique à ceux-ci. Il est cependant possible que des avancés autant théoriques que pratiques permettent de tenir un vote par Internet de

façon sécuritaire dans le futur. Il est malheureusement impossible de prédire à quel moment cela sera le cas.

Résumé des enjeux

Difficultés actuelles à développer du logiciel

Le manque de maturité pour l'identification de vulnérabilités fait en sorte que l'on ne peut pas avoir un niveau raisonnable de confiance qu'un produit est absent de vulnérabilités. Ceci affecte potentiellement la résilience à l'interférence et l'anonymat des votes.

Le manque de maturité pour l'identification de bogue fonctionnel fait en sorte que l'on ne peut pas avoir un niveau raisonnable de confiance qu'un produit est absent de bogues. Ceci affecte potentiellement l'accessibilité du système de votation.

Installations domestiques non adaptées et/ou difficiles à supporter

Le matériel informatique dont le citoyen moyen dispose à la maison répond difficilement aux requis de sécurité et de disponibilité que l'on devrait avoir. Ceci affecte potentiellement l'anonymat des votes et la résilience à l'interférence.

La très grande variété d'appareils et de configurations rend très difficile le développement d'une application web qui serait accessible à l'ensemble de la population. Ceci affecte potentiellement l'accessibilité du système de votation.

Angle de compromission très large

La question de la sécurité d'un système de votation par Internet dépasse largement la vérification de l'absence de vulnérabilités. Beaucoup de cas vécus existent pour le démontrer. Ceci affecte potentiellement l'anonymat des votes et la résilience à l'interférence.

Transparence difficile à obtenir pour les citoyens

Le développement d'un système de votation transparent est difficile à faire en pratique et aucun pays n'a été capable d'en mettre un en place qui est transparent. Ceci affecte potentiellement la résilience à l'interférence.

Enjeux

Difficultés actuelles à développer du logiciel

Manque de maturité pour l'identification de vulnérabilités

Le domaine de l'identification de vulnérabilité dans du logiciel est encore à ses débuts et il est très difficile de qualifier ce domaine de mature. Pour bien démontrer ce point, les éléments suivants sont à considérer.

Délai d'identification des vulnérabilités

Pour que l'on considère le domaine mature, on devrait s'attendre à ce que des vulnérabilités majeures dans des produits ou bibliothèques logicielle très largement utilisés (ex.: OpenSSL) soient identifiées dans un délai raisonnable. Ces produits ou bibliothèques logicielle font souvent l'objet de revues de sécurité. Si l'on est incapable de le faire pour des bibliothèques logicielle et/ou produits qui sont largement utilisés, comment peut-on s'attendre à être capable de le faire pour un logiciel qui disposera d'encore moins de ressources ?

Il existe beaucoup d'exemples de vulnérabilité majeure qui ont existé pendant plusieurs années dans des produits ou bibliothèques logicielle avant d'être identifiés. En voici quelques cas de figure notable :

- Heartbleed[4] est la pire vulnérabilité à être identifiée dans la bibliothèque logicielle OpenSSL et est souvent nommée comme une des pires vulnérabilités à avoir été identifiée dans une bibliothèque logicielle. Un délai de près de deux ans existe entre l'introduction de la vulnérabilité et sa découverte.
- La vulnérabilité Shellshock[5] a été introduite dans l'utilitaire bash en 1989 et n'a été découverte et corrigée qu'en 2014.
- La vulnérabilité DROWN[6] publiée en 2016 affecte des installations qui supportent à la fois SSLv2 et d'autres versions de SSL. Ce type d'installation était très présent sur Internet depuis autour de 1996 (date de sortie de SSLv3).

Constante découverte de nouvelles classes de vulnérabilité

On ne peut pas avoir un niveau de confiance acceptable qu'un logiciel est absent de vulnérabilité s'il n'existe pas une liste finie de classes de vulnérabilité. Comment peut-on être certain qu'on a tout identifié, si l'on ne connait même pas l'ensemble des classes de vulnérabilité ? À ce jour, on est très loin de pouvoir considérer les classes de vulnérabilité comme étant complètement connues. Voici quelques

[4] http://heartbleed.com/
[5] https://en.wikipedia.org/wiki/Bash_(Unix_shell)
[6] https://drownattack.com/

points qui démontrent bien cette réalité :

- Depuis la dernière année, 92 nouvelles classifications de vulnérabilités ont été ajoutées au corpus CWE (Common Weakness Enumerations)[7].

- Depuis 2 ans, PortSwigger[8][9] publie chaque année un palmarès des meilleures nouvelles recherches dans le domaine de la sécurité applicative web. Parmi ces recherches, on trouve chaque année de nouvelles classes de vulnérabilité spécifiques aux applications web.

Absence de méthodologie de test

En ce moment, il n'existe pas vraiment de méthodologie largement acceptée sur comment s'y prendre pour identifier des vulnérabilités. Ce qui se rapproche le plus d'une méthodologie de test dans le domaine est l'utilisation de liste de vérification (checklist). Plusieurs projets très connus comme ASVS[10] et OWASP Testing Guide[11] sont de bons points de départ. Par contre, les listes de vérification ont plusieurs lacunes majeurs lorsque l'on cherche à identifier l'ensemble des vulnérabilités. L'écriture de ces documents est longue et il peut s'écouler plusieurs années entre chacunes de leurs mises à jour. Ainsi, toute classe de vulnérabilité identifiée depuis la dernière mise à jour ne sera pas présente. Plusieurs types de vulnérabilité, comme des bogues de logique, ne sont pas présents dans ces listes. Ceci fait en sorte qu'actuellement le résultat d'un audit de code professionnel peut donner des résultats très variables.

Méthode plus rigoureuse d'analyse limitée ou pas encore mature

Des méthodes d'analyse formelle[12] commencent à être utilisées dans certains domaines très précis pour l'identification de vulnérabilités ou constater l'absence de certaines classes de vulnérabilité. Même si ces techniques sont très prometteuses, car elles permettent d'avoir une approche systématique et rigoureuse face à l'identification de vulnérabilités, il est important de mettre certains points au clair :

- On ne peut pas faire de l'analyse formelle sans avoir une définition mathématique de ce qu'on doit prouver. La définition de ce que c'est une vulnérabilité est beaucoup trop imprécise pour qu'on puisse penser faire de l'analyse formelle qui prouverait l'absence de vulnérabilités. L'analyse formelle sert au mieux à prouver qu'une classe de vulnérabilité est présente ou pas.

- Certaines classes de vulnérabilité ne peuvent pas être identifiées avec de l'analyse formelle[13] [14].

---

[7] https://cwe.mitre.org/news/index.html
[8] https://portswigger.net/research/top-10-web-hacking-techniques-of-2017
[9] https://portswigger.net/research/top-10-web-hacking-techniques-of-2018
[10] https://www.owasp.org/index.php/Category:OWASP_Application_Security_Verification_Standard_Project
[11] https://www.owasp.org/index.php/OWASP_Testing_Project
[12] https://en.wikipedia.org/wiki/Formal_methods
[13] https://defiprime.com/defi-smart-contract-audits#dan-guido-trail-of-bits-5

- Ce type de méthode est encore très nouveau et il existe très peu d'expertise sur ce sujet.

Manque de maturité pour l'identification de bogue fonctionnel

Dans les dernières années, il n'est pas très difficile de trouver des échecs importants dans le développement et le déploiement de logiciels. On n'a qu'à penser au système de paie Phénix[15] qui a été un échec monumental.

Au-delà de ces échecs, il faut aussi regarder ce qui est actuellement considéré comme étant l'état de l'art dans le domaine. Pour s'assurer qu'un logiciel répond aux besoins fonctionnels et est absent de bogues, on utilise ce qu'on appelle des tests. Un test valide qu'une portion du logiciel réagit selon la spécification qui a été faite. On mesure généralement la complétude de ces tests avec des métriques comme la couverture de code et la couverture de branche. La principale limitation de cette approche est qu'elle reste approximative. Même avec une couverture de code et une couverture de branche à 100%, il peut toujours subsister des bogues. Donc, même si cette approche est efficace pour l'identification de bogues, elle ne peut pas garantir l'absence de ceux-ci.

Il faut aussi considérer qu'au-delà des bogues fonctionnels, il peut aussi y avoir des bogues dans l'affichage du contenu dans le navigateur. S'assurer que le rendu d'une interface utilisateur dans un navigateur web est fait correctement est encore très difficile à faire.

Difficulté à spécifier des requis de sécurité et des requis non fonctionnel

Pour une application web de votation en ligne, les requis non fonctionnels (la sécurité de l'application web, l'accessibilité, la facilité d'utilisation, etc.) sont une partie très importante de l'application web. Comme mentionné précédemment, il n'existe pas de méthodologie complète pour valider qu'un logiciel est absent de vulnérabilités. Ceci pose un problème important au niveau de la spécification des requis de sécurité. Toute spécification qui stipule que le logiciel livré doit être absent de vulnérabilités est impossible à valider.

Lorsqu'un produit est développé en impartition, le contractant fait souvent le minimum qui respecte les exigences vérifiables du contrat afin de maximiser son profit. Il est donc difficile d'envisager qu'une telle application web développée en impartition soit réellement sécuritaire et sans bogue. La même problématique existe dans une moindre mesure lorsque l'application web n'est pas développée en impartition. Les gestionnaires de projet et les développeurs font souvent face à de la pression pour livrer un produit qui n'est pas encore prêt. S'il n'est pas possible de valider les requis non fonctionnels, il n'est pas possible de savoir si l'application web est réellement prête.

[14]https://blog.trailofbits.com/2019/08/08/246-findings-from-our-smart-contract-audits-an-executive-summary/

[15] https://ici.radio-canada.ca/sujet/phenix-2018

Installations domestiques non adaptées et/ou difficiles à supporter

Sécurité des ordinateurs personnels

Les ordinateurs personnels n'ont jamais été conçus pour avoir un niveau de sécurité adéquat pour le vote électronique. La population moyenne ne garde pas toujours ses logiciels à jour et installe des logiciels d'un peu n'importe où sur Internet. Il est donc fréquent que des ordinateurs personnels se fassent infecter par des logiciels malveillants.

Au-delà des logiciels malveillants, il faut aussi considérer que des logiciels de type 'stalkerware'[16] sont malheureusement de plus en plus présents et utilisés. Ces logiciels permettent d'espionner l'appareil mobile ou l'ordinateur de la victime. Ils sont généralement installés par quelqu'un qui est proche de la victime à son insu. Ce genre de logiciel pourrait être utilisé pour voir le vote d'une autre personne et ainsi brimer son droit au vote secret.

On peut donc difficilement dans ce contexte considérer les ordinateurs ou téléphones personnels comme adéquats pour voter.

Vote à partir du bureau

On doit prévoir que beaucoup de citoyens vont voter à partir de leur lieu de travail. Les grandes compagnies mettent souvent en place des outils d'interception SSL[17] dans leur réseau pour des raisons de sécurité. Ces outils sont capables de voir en clair toutes les requêtes qui sont faites à des applications web. Un employeur pourrait donc avoir accès aux votes de ses employés.

Variété importante de navigateur web, configuration et extensions

Même si le développement web s'est simplifié avec les années puisque les différences fonctionnelles entre les principaux navigateurs web ont beaucoup diminué, il est encore difficile de supporter la quasi-totalité des installations domestiques. L'approche que la très grande majorité des compagnies de développement logiciel optent face à ce problème est de supporter ce qui est rentable de supporter et d'ignorer le reste. Cette approche est viable pour une compagnie commerciale. Par contre, dans le contexte d'une application web de vote par Internet, il n'est pas vraiment acceptable de dire que l'on va supporter uniquement 95%, 98% ou 99% des installations domestiques.

Il est bon de noter que les citoyens qui n'ont pas accès à des ordinateurs à jour font généralement parties des segments plus pauvre de la population. Choisir de supporter uniquement les installations récentes affecteraient disproportionnement la capacité de voter de ces segments de la population.

[16] https://www.vice.com/en_us/article/53vm7n/inside-stalkerware-surveillance-market-flexispy-retina-x
[17] https://docs.citrix.com/en-us/citrix-adc/13/forward-proxy/ssl-interception.html

Pour qu'une application web soit fonctionnelle sur la quasi-totalité des installations domestiques, les points suivants sont à considérer :

- Un très large éventail de versions de navigateur web, incluant des versions désuètes, doit être supporté. Une partie de la population ne possède pas d'ordinateur très puissant ou récent. Il n'est donc pas rare qu'il utilise de plus vieilles versions de navigateur. De plus, ce n'est pas toute la population qui garde ses logiciels à jour. Ce qui rend aussi la situation difficile est qu'il est fréquent que les installateurs pour les plus vieilles versions de navigateur ne soient plus disponibles. Il devient donc difficile dans ce contexte de tester adéquatement l'application web.

- L'utilisation d'extensions pour les navigateurs web est de plus en plus fréquente. Certaines extensions populaires (ex.: AdBlock, uBlock Origin, etc.) modifient le contenu retourné par les sites web. Ceci peut empêcher certaines fonctionnalités de fonctionner correctement. Pour atteindre un large support, il faut donc s'assurer que l'application web reste fonctionnelle avec la totalité des extensions les plus populaires. Un autre point à considérer concernant les extensions est que certaines d'entre elles sont utilisées pour des raisons d'accessibilité. Il n'est donc pas acceptable de demander de les désactiver pour utiliser l'application web de vote en ligne.

- Les navigateurs possèdent plusieurs configurations qui affectent l'affichage et les fonctionnalités disponibles. Plusieurs de ces configurations (ex.: taille du texte) sont utilisées pour des raisons d'accessibilité. Il faut donc que l'application web soit fonctionnelle avec l'ensemble des configurations d'accessibilité des différents navigateurs.

Être capable de respecter l'ensemble de ces points est en pratique ardu. Ce sont des points qui peuvent être facilement oubliés et rendre les coûts et temps de développement très élevés.

Angle de compromission très large

Un des points que l'on cherche à mettre en valeur dans ce mémoire est que la protection d'une application web de vote par Internet ne passe pas uniquement par des analyses de sécurité sur l'application web elle-même. Un attaquant peut compromettre l'application web de vote par Internet sans avoir à exploiter de vulnérabilités dans celle-ci.

Il est bon de noter que plus les enjeux sont importants, plus les acteurs qui vont essayer d'attaquer le système vont avoir des moyens monétaires et des connaissances techniques. Ce qui est présenté dans cette section est applicable pour une application web de vote par Internet pour des élections provinciales ou fédérales, mais n'est pas vraiment applicable si l'enjeux était, par exemple, un vote de grève pour une association étudiante.

Attaque par la chaîne d'approvisionnement (supply-chain attack)

Les attaques par chaîne d'approvisionnement consistent à compromettre une application web en compromettant soit la compagnie qui la développe, soit les mécanismes de distributions des dépendances ou encore les développeurs des dépendances afin que le code de l'application web contienne une porte dérobée (backdoor). Les cas de ce type d'attaque se sont multipliés dans les dernières années et ne peuvent plus être ignorés. Plusieurs cas majeurs sont pertinents à souligner pour mieux comprendre cet enjeu.

Cas élection américaine

Le cas le plus pertinent à l'enjeu du vote par Internet concerne l'interférence de la Russie lors de la dernière campagne électorale américaine. On apprend dans le rapport Mueller[18] que la Russie a ciblé des compagnies qui développent des composants pour le vote électronique et aurait infecté leur réseau. L'objectif exact de l'attaque n'est pas indiqué dans le rapport (c'est expliqué dans le rapport que ce point était essentiellement hors portée). On peut tout de même constater que la Russie a perçu ces compagnies comme des cibles pertinentes.

« The GRU also targeted private technology firms responsible for manufacturing and administering election-related software and hardware, such as voter registration software and electronic polling stations »

(page 50)

« Unit 74455 also sent spearphishing emails to public officials involved in election administration and personnel at companies involved in voting technology. In August 2016, GRU officers targeted employees of , a voting technology company that developed software used by numerous U.S. counties to manage voter rolls, and installed malware on the company network »

[18] https://www.justice.gov/storage/report.pdf

(page 51)

Ce qui est important à retenir, c'est que l'équipe qui développe, maintient et gère l'application web ou l'infrastructure de vote par Internet doit être capable de détecter et résister à une attaque d'un état nation.

Cas Magecart

Le cas d'un groupe d'attaquants nommé Magecart[19] est aussi intéressant à étudier. Ce groupe cherche principalement à voler des informations de cartes de crédit et autres informations bancaires. Le modus operandi pour compromettre des sites où il y avait du paiement en ligne est le suivant :

- Le groupe identifie des ressources hébergées par des tierces parties qui sont incluses dans les pages de paiement de sites de commerce électronique.
- Le groupe compromet les tierces parties en question et modifie les ressources qui sont servies aux sites de paiement en ligne.
- Quand une personne arrive sur la page de paiement en question, la page contient du code malicieux qui provient d'une tierce partie. Ce code malicieux capture les informations de paiement qui sont saisies par l'utilisateur et les transmet à un service contrôlé par le groupe Magecart.

Ce qui est intéressant avec le modus operandi de ce groupe est que les sites de commerce électronique ne sont pas directement compromis. La compromission passe entièrement par des tierces parties que le site utilise. Ce qui est important à retenir, c'est que non seulement la plateforme de votation par Internet doit être sécure, mais aussi l'ensemble des services qu'elle utilisera.

Cas CoPay

CoPay est une application de gestion de cryptomonnaie. En 2018, une des bibliothèques logicielle que l'application utilise est compromise et l'attaquant publie une nouvelle version de la bibliothèque logicielle avec une porte dérobée qui permet de voler la cryptomonnaie contenue dans les portefeuilles gérés par l'application CoPay[20]. La nouvelle version de la bibliothèque logicielle compromise est intégrée à l'application de CoPay. Un nombre inconnu d'utilisateurs téléchargeront la version compromise de l'application et se feront voler leur cryptomonnaie. La compromission s'est faite ici en s'attaquant aux bibliothèques logicielle que l'application utilisait et non à l'application elle-même.

Il est pratiquement impossible de développer quoi que ce soit comme application web moderne sans utiliser de bibliothèques logicielle. Ces bibliothèques logicielle peuvent être une source compromission même quand elles sont maintenues par des entités fiables et reconnues. Être certain que les

[19]https://blog.trendmicro.com/trendlabs-security-intelligence/new-magecart-attack-delivered-through-compromised-advertising-supply-chain/

[20] https://www.zdnet.com/article/hacker-backdoors-popular-javascript-library-to-steal-bitcoin-funds/

bibliothèques logicielle que l'on utilise ne contiennent pas de porte dérobée ou vulnérabilités intentionnelles est excessivement difficile, voire impossible, à déterminer en pratique.

Chaîne d'approvisionnement physique

Un aspect important à considérer lors de la conception d'un système de votation par Internet est comment la transmission d'identifiant est faite. Cette conception doit faire en sorte qu'il ne puisse pas y avoir d'acteur capable de récupérer les informations d'identification de beaucoup de citoyens. C'est en pratique difficile, car les moyens de transmissions traditionnels comme la poste passent tous par un nombre limité de personnes. Un facteur qui, par exemple, volerait les envois d'Élection Québec pour le vote par Internet pourrait voter en tant que les citoyens de son secteur. C'est le genre d'attaque qu'il faut prévoir dans la conception du système.

Prévalence et efficacité de l'hameçonnage (phishing)

L'hameçonnage est un problème duquel il est fondamentalement difficile de se protéger pour les applications web. L'hameçonnage fonctionne généralement comme ceci : la personne mal intentionnée héberge une copie du vrai site web sur de l'infrastructure qui n'appartient pas à l'entité qu'il attaque et il envoie des messages incitant à visiter le faux site et fournir des informations par des services qui n'appartiennent pas à l'entité qu'il attaque. Comme l'entité ciblée n'a aucun contrôle sur ces deux aspects, il est toujours difficile pour elle de rapidement mettre fin à ce genre de campagne d'hameçonnage.

Qu'est-ce qu'il arrivera si quelqu'un envoie une campagne d'hameçonnage le jour de l'élection pour obtenir les informations d'identification fournie par Élections Québec pour voter en ligne ? Même si ce genre d'attaque n'est pas subtil à faire en pratique, il pourrait sérieusement mettre en doute la validité d'un résultat électoral si les autorités ne sont pas adéquatement préparées.

La problématique n'est pas impossible à gérer, mais requiert une bonne préparation. Il serait recommandé qu'Élections Québec consulte des experts qui travaillent sur cette problématique. Certains secteurs d'affaire comme les secteurs financiers ont d'ailleurs une bonne expertise sur ce sujet.

Déni de service (DOS et DDOS) difficile à mitiger

Les attaques de déni de service consistent généralement à générer un énorme volume de trafic vers un serveur précis. Lors de ce type d'attaque, des requêtes faites par des utilisateurs légitimes ne sont plus capables d'être traitées par le serveur. La seule réelle façon de se prémunir de ce genre d'attaque est d'avoir une connexion Internet avec plus de capacité que ce que l'attaquant est capable de générer comme trafic. L'attaque de type DDoS la plus importante à ce jour est celle qui a visé le service GitHub en 2018. L'attaque à son plus fort générait 1.35Tbps de trafic vers les serveurs de GitHub[21]. Il est bon de noter qu'aucun fournisseur de service canadien ne peut offrir une connexion Internet suffisante pour mitiger une attaque de cette ampleur.

[21] https://github.blog/2018-03-01-ddos-incident-report/

L’approche qui est souvent utilisée en entreprise pour pallier à ce problème est l’utilisation de service de protection DDoS (ex.: Cloudflare). Il y a cependant plusieurs enjeux importants à l’utilisation de ces services :

- Ces services ont besoin de faire de la terminaison SSL pour fonctionner. Ceci implique que le service de protection DDoS a accès en clair à tout le trafic qui circule vers le serveur final. C’est une caractéristique qui n’est pas désirable puisqu’elle pourrait affecter la confidentialité des votes.
- Les principales compagnies qui offrent ce type de service sont toutes américaines. Ceci implique que l’ensemble du trafic lié au vote par Internet passerait par de l’infrastructure étrangère.
- Ces services utilisent souvent des techniques qui sont peu accessibles lorsque le site en question est ciblé par une attaque de déni de service. Ce genre d’approche a été critiqué dans le passé[22] [23] .

Dépendance aux compagnies étrangères

Un aspect qu’il est important de regarder concernant l’ingérence possible de pays étrangers dans le processus électoral est l'utilisation de services opérés ou développés par des compagnies situées en pays étranger. En ce moment ce sont les États-Unis qui dominent le marché du développement de logiciel, d’infrastructure en ligne et de services en ligne. Il est actuellement ardu de développer un produit en n’utilisant aucun produit sous contrôle de compagnie étrangère. Même si l’enjeu n’est pas très important pour l’instant (le Canada a de bonnes relations diplomatiques avec la plupart des autres pays) il faut considérer que ce point peut changer dans le futur et pourrait devenir problématique.

En dehors des logiciels et services, il faut aussi considérer que la quasi-totalité du matériel physique (ex.: CPU, mémoire RAM, etc.) est actuellement fabriquée en Asie. Ce type de matériel est actuellement impossible à inspecter pour s’assurer qu’il n’y a pas de porte dérobée.

[22] https://proprivacy.com/privacy-news/cloudflare-recaptcha-nightmare
[23] https://www.gov.uk/service-manual/technology/using-captchas

Transparence difficile à obtenir pour les citoyens

La transparence du processus électoral est un aspect fondamental à conserver. Il est difficile de maintenir un lien de confiance dans le processus électoral lorsque ce processus est opaque ou que des composants majeurs doivent être gardés secrets.

Pour que le processus électoral reste transparent, les éléments suivants doivent être considérés.

- Comment l'installation physique des serveurs a été faite doit être public et vérifiable.
- Des personnes qualifiées doivent faire une validation indépendante des composants physiques installés pour détecter la présence de porte dérobée.
- Quel système d'exploitation est utilisé sur les serveurs et comment il a été installé doit être public et vérifiable.
- Le code source du système de votation doit être public.
- Le déploiement du code source du système de votation doit être vérifiable. On doit pouvoir être capable de valider si le code source déployé est exactement le bon.

Actuellement, il existe peu ou pas d'expertise sur la mise en place de système qui requiert ce niveau de transparence. Il n'existe à notre connaissance aucun pays qui a été capable de mettre en place un système de votation en ligne ou électronique qui est complètement transparent. Les problématiques qui sont mentionnées sont en pratique ardues à respecter. Il faut s'attendre à ce que cela prenne du temps avant qu'on soit capable de développer et mettre en place des systèmes qui sont complètement transparents.

Mythes

Il est pertinent d'adresser certains mythes concernant le vote par Internet. Il est probable qu'ils soient présentés durant les consultations publiques et on tenait à ce qu'ils soient démentis.

Comparaison avec les systèmes bancaires ou application de commerces électroniques

Beaucoup de gens sont portés à croire que parce que l'on a des systèmes bancaires et des applications de commerce électronique fonctionnels que l'on doit nécessairement être capable de mettre en place un système de votation sécuritaire. Cette comparaison est malheureusement erronée.

Il faut tout d'abord comprendre que les contraintes dans lesquelles évoluent les système bancaires sont fondamentalement différentes de celle des systèmes de votation électronique. Les systèmes bancaires conservent des traces de toutes les transactions effectuées, alors que les systèmes de votation électronique ne peuvent pas conserver ce genre d'information pour des questions d'anonymat.

Il faut aussi comprendre que les systèmes bancaires et les applications de commerces électroniques sont loin d'être complètement sécuritaires. Aucune compagnie de grande taille dans ce domaine à des pertes nulles lié à la fraude ou aux brèches de sécurité. À titre indicatif, Interac publie chaque année le volume de fraude qu'elle subit. L'an dernier, son volume de fraude était de 4.4 millions de dollars[24]. De plus, en 2017, la Banque du Canada a publié un rapport dans lequel il est indiqué que les banques canadiennes perdent environ 800 millions de dollars annuellement en fraude par carte de crédit[25].

Blockchain

Un des plus persistants est que l'utilisation du "blockchain" règlerait tous les enjeux de sécurité. Ce qu'il faut comprendre est que non seulement le "blockchain" n'adresse pas les principaux enjeux de sécurité liés au vote par Internet, mais en plus introduit des faiblesses au niveau de la confidentialité du vote. Fondamentalement, le "blockchain" est une structure de données dans laquelle des données sont ajoutées séquentiellement de façon immuable. Si un groupe de personnes connaît les votes qu'ils ont faits et qu'ils sont capables d'identifier où dans le "blockchain" cette séquence de vote apparaît, ils sont aussi capables de déterminer les votes qui ont été faits juste avant ou après. Ainsi, le vote de toute personne qu'ils connaissent qui a voté autour du même moment qu'eux n'est plus secret. Ce type de stockage est donc inapproprié pour le vote électronique.

[24] https://www.interac.ca/en/fraud.html
[25] https://www.bankofcanada.ca/wp-content/uploads/2018/12/sdp2018-17.pdf

David Jefferson a écrit un bon papier sur ce mythe qu'il est fortement recommandé de consulter en complément d'information[26].

PCI

PCI est un standard de conformité qui est présent et utilisé par les compagnies qui font du paiement par carte de crédit. Il est malheureusement courant d'entendre des affirmations qui insinuent que parce qu'une compagnie est conforme à PCI qu'elle doit nécessairement être sécuritaire. Pour comprendre pourquoi cela est faux, il faut tout d'abord comprendre ce qu'un standard de conformité est et n'est pas. Un standard de conformité est fondamentalement une liste de choses à faire et d'exigences à respecter. Ces listes ne peuvent pas être exhaustives pour les raisons mentionnées dans la section "Manque de maturité du domaine pour l'identification de vulnérabilités". La bonne façon de voir ces standards est qu'il s'agit d'un minimum qui devrait être fait.

Le standard PCI a aussi plusieurs problèmes qui sont intrinsèquement liées à son fonctionnement. Le premier est que le standard n'est que très rarement mis à jour. On y retrouve actuellement des aberrations qui ne sont pas encore à jour. Le standard recommande entre autres de changer régulièrement les mots de passe des employés[27], alors que cette pratique est fortement déconseillée par la dernière révision du standard NIST sur la gestion des mots de passe[28]. La dernière révision du glossaire de PCI DSS[29] mentionne encore que TDES est considéré comme de la cryptographie forte, alors que des faiblesses ont été démontrées avec cet algorithme[30]. Le deuxième est dans la vérification du respect du standard. Même lorsque des audits PCI externes sont faits, les auditeurs de PCI se basent en très grande partie sur la documentation fournie par la compagnie qui se fait auditer pour prendre la décision d'octroyer la certification ou non. La problématique avec cet enjeu est qu'il est facile pour une compagnie de cacher des non-respects au standard, parce qu'elle peut présenter les documents de son choix.

Il existe une panoplie d'autres standards de conformité dans l'industrie et la plupart des points soulevés par rapport à PCI sont aussi applicables à ces autres standards. On ne peut donc pas conclure que parce qu'une compagnie est certifiée pour un standard de l'industrie qu'elle est nécessairement sécuritaire.

[26]https://www.verifiedvoting.org/wp-content/uploads/2018/10/The-Myth-of-_Secure_-Blockchain-Voting-1002.pdf
[27]https://www.pcisecuritystandards.org/documents/Payment-Data-Security-Essential-Strong-Passwords.pdf?agreement=true&time=1572118363193
[28] https://pages.nist.gov/800-63-3/sp800-63b.html
[29]https://www.pcisecuritystandards.org/documents/PCI_DSS_Glossary_v3-2.pdf?agreement=true&time=1572118679043
[30] https://sweet32.info/

Recommandations

On trouve pertinent de faire des recommandations sur ce qui permettrait d'éventuellement mettre en place une application web de vote par Internet. Ce sont toutes des recommandations que l'on considère importantes pour que les enjeux mentionnés dans ce mémoire soient éventuellement adressés.

Code source public

Le code source source de l'ensemble de l'application web de vote par Internet devrait être public. Ceci veut dire que le public devrait avoir accès au code source de l'application web ainsi que tout ce qui est utilisé comme scripts, système d'exploitation, etc. pour l'installation de l'application web sur les serveurs. Ceci permet d'adresser les enjeux de transparence et mitige en partie les enjeux liés aux attaques par chaîne de compromission.

Développement interne

Il est très fortement déconseillé que le développement du système de votation par Internet soit fait par une tierce partie. L'expertise pour le développement de ce type de logiciel n'existe pas ou très peu au Québec. Il n'y a donc aucun avantage à faire développer le logiciel par une compagnie privée ou par des consultants. Élection Québec a avantage à long terme à se développer une expertise interne dans ce domaine.

Utilisation limitée

On reconnaît que l'expertise dans le développement et l'opérationnalisation de ce type d'application web ne se fait pas toute seule et que la meilleure façon d'apprendre est souvent d'essayer. On recommande que si Élection Québec veut faire des essais pour se bâtir une expertise qu'elle se limite à des élections de moindre envergure (ex.: élection dans des petites municipalités) ou qu'elle limite l'utilisation du vote par Internet à un pourcentage peu significatif du vote (ex.: moins que 0.5%). Il est fortement recommandé de respecter cette recommandation jusqu'à ce qu'un niveau acceptable de confiance envers le système soit atteint.

Modèle confiance

Même s'il est impossible d'avoir une garantie à 100% qu'un logiciel est absent de vulnérabilités, il est tout de même possible d'atteindre un point où on a un niveau de confiance raisonnable qu'il n'y a pas de vulnérabilités majeures. Pour atteindre ce point, il est fortement recommandé de s'inspirer de la méthodologie derrière le choix des algorithmes cryptographiques (ex.: AES, SHA3, etc.)[31]. Ce processus peut être vu à haut niveau en 5 grandes phases.

[31] https://en.wikipedia.org/wiki/Advanced_Encryption_Standard_process

- Au départ, des chercheurs soumettent des propositions d'algorithme.
- Ces algorithmes sont ensuite publiés et publiquement discutés.
- Une sélection est ensuite faite par des experts du domaine en se basant sur les discussions publiques afin de sélectionner ce qui est probablement la meilleure option.
- Il se déroule ensuite une période de un ou deux ans pendant laquelle l'algorithme sélectionné est étudié.
- Si aucune attaque significative n'est trouvée, la sélection est considérée concluante et l'algorithme commence à être utilisé à grande échelle.

C'est un processus qui s'étire sur plusieurs années, mais qui permet d'avoir un bon niveau de confiance envers le choix qui a été fait malgré qu'il soit impossible de prouver qu'un algorithme est absent de faiblesse importante.

Cas d'étude complémentaire

L'Estonie utilise le vote par Internet depuis 2005 et plusieurs analyses indépendantes ont été faites de ce système de votation. Il est fortement recommandé de consulter ces deux analyses/papiers qui ont été produits par des équipes indépendantes en 2014 et 2011 au sujet du système de vote par Internet de l'Estonie. Ces analyses/papiers mettent en lumière les faiblesses actuelles du système de vote par Internet de l'Estonie et des difficultés rencontrées lors de sa mise en place. La conclusion du papier "Security Analysis of the Estonian Internet Voting System" est d'ailleurs particulièrement critique face à la possibilité de pouvoir faire du vote par Internet sécuritairement. Il est entre autres mentionné que les problèmes fondamentaux liés au vote par Internet ne sont toujours pas réglés. Dans leur conclusion, les auteurs recommandent d'ailleurs de discontinuer le système de vote par Internet de l'Estonie.

Security Analysis of the Estonian Internet Voting System, 2014
https://dl.acm.org/citation.cfm?id=2660315

Report on the Estonian Internet Voting System, 2011
https://www.verifiedvoting.org/report-on-the-estonian-internet-voting-system-2/

Introduction

Ce document présente les personnes qui appuient le mémoire intitulé “Mémoire sur le vote par Internet” écrit par “Olivier Arteau” et soumis le “3 novembre 2019”. Ce sont tous des professionnels qui travaillent dans de le domaine du développement logiciel ou de la sécurité informatique à Montréal.

Signataires

Jean Privat, professeur à l'UQAM au département d'informatique et directeur de programmes

Jean-Sébastien Fauteux, B. Ing

Kevin Carroll, Consultant en cybersécurité

Laurent Desaulniers, Directeur Test d'intrusion

Olivier Bilodeau, Chercheur en cybersécurité