Le bulletin de vote est annulé avant la blockchain : une analyse de la sécurité de Voatz, la première application de vote par Internet utilisée lors des élections fédérales américaines

Michael A. Specter, James Koppel et Daniel Weitzner, MIT

https://www.usenix.org/conference/usenixsecurity20/presentation/spectre

Cet article est inclus dans les Actes du 29e USENIX Security Symposium.

Du 12 au 14 août 2020

978-1-939133-17-5

Le bulletin de vote est bloqué avant la blockchain : une analyse de la sécurité de Voatz, la première application de vote par Internet utilisée aux États-Unis Élections fédérales

Michael A. Spectre
MIT †

James Koppel
MIT‡

Daniel Weitzner
MIT§

Abstrait

Lors des élections de mi-mandat de 2018, la Virginie-Occidentale est devenue le premier État des États-Unis à autoriser certains électeurs à voter sur un téléphone mobile via une application propriétaire appelée "Voatz". Bien qu'il n'y ait pas de description formelle publique du modèle de sécurité de Voatz, la société affirme que la sécurité et l'intégrité des élections sont maintenues grâce à l'utilisation d'une blockchain autorisée, de la biométrie, d'un mixnet et de modules de stockage de clés basés sur le matériel sur l'appareil de l'utilisateur. Dans ce travail, nous présentons la première analyse de sécurité publique de Voatz, basée sur une ingénierie inverse de leur application Android et la documentation minimale disponible du système. Nous avons effectué une réimplémentation en salle blanche du serveur de Voatz et présenté une analyse du processus électoral tel qu'il est visible depuis l'application elle-même.

Nous constatons que Voatz présente des vulnérabilités qui permettent à différents types d'adversaires de modifier, d'arrêter ou d'exposer le vote d'un utilisateur, y compris une attaque par canal latéral dans laquelle un adversaire de réseau complètement passif peut potentiellement récupérer le lot secret d'un utilisateur . Nous constatons en outre que Voatz a un certain nombre de problèmes de confidentialité résultant de son utilisation de services tiers pour des fonctionnalités cruciales de l'application. Nos résultats servent d'illustration concrète de la sagesse commune contre le vote par Internet et de l'importance de la transparence pour la légitimité des élections. À la suite de notre travail, un comté de Washington a déjà interrompu son utilisation de Voatz lors des primaires de 2020.

1. Introduction

En 2018, Voatz, une société privée basée à Boston, est entrée dans l'histoire en proposant la première application de vote par Internet utilisée dans les jeux à enjeux élevés.[1] Élections fédérales américaines. Ciblant principalement les militaires étrangers et d'autres électeurs absents, Voatz a été utilisé lors des élections fédérales, étatiques et municipales en Virginie-Occidentale, à Denver, en Oregon et en Utah, ainsi qu'à la Convention démocratique du Massachusetts de 2016 et à la Convention républicaine de l'Utah de 2016 [45] . La société a récemment clôturé une série A de 7 millions de dollars [27] et est sur la bonne voie pour être utilisée dans les primaires de 2020.

Dans cet article, nous présentons la première revue de sécurité publique de Voatz. Nous constatons que Voatz est vulnérable à un certain nombre d'attaques qui pourraient violer l'intégrité des élections (résumé dans le tableau 1). Par exemple, nous constatons qu'un attaquant ayant un accès root à l' appareil d'un électeur peut facilement échapper aux défenses du système (§5.1.1), apprendre les choix de l'utilisateur (même après la fin de l'événement) et modifier le vote de l'utilisateur (§5.1) . Nous constatons en outre que leur protocole réseau peut divulguer les détails du vote de l'utilisateur (§5.3) et, étonnamment, que l'utilisation de la blockchain par le système est peu susceptible de protéger contre les attaques côté serveur (§5.2). Nous fournissons une analyse de ces défauts et constatons que l'exploitation serait tout à fait à la portée d'un acteur de l'État-nation.

Bien que l'introduction du vote par Internet aux États-Unis soit relativement nouvelle, l'histoire entourant le vote électronique uniquement ne l'est pas. À la suite d'erreurs de comptage, d'écarts de recomptage et de bulletins de vote ininterprétables qui ont fait des ravages lors de la course présidentielle américaine de 2000, le Congrès a adopté le Help America Vote Act (HAVA) [59], un projet de loi visant à aider les États à s'éloigner des cartes perforées obsolètes et problématiques. systèmes basés . La Commission d'assistance électorale (EAC), une nouvelle agence exécutive créée par la HAVA, a été chargée de distribuer ces fonds et a depuis fourni plus de 3,3 milliards de dollars à divers États pour aider à améliorer l'infrastructure électorale [31].

Malheureusement, HAVA ne disposait pas de directives strictes sur les systèmes de remplacement autorisés à être achetés. En conséquence, de nombreux États ont acquis des machines à voter exclusivement électroniques non contrôlées , connues sous le nom de systèmes d'enregistrement électronique direct (DRE). De nombreuses études ont depuis montré que les systèmes DRE sont extrêmement vulnérables à un large éventail d'attaques, permettant aux adversaires de modifier subrepticement le résultat d'une élection [21, 22, 33, 49, 77].

Aujourd'hui, nous assistons à des développements similaires en réponse

---

Avec notre gratitude à Barbara Simons [46]

†Candidat au doctorat EECS, CSAIL, Internet Policy Research Initiative
‡Candidat au doctorat EECS, CSAIL, Computer Assisted Programming Group
§Research Scientist, CSAIL, Internet Policy Research Initiative

[1] où les adversaires sont susceptibles de dépenser des ressources pour modifier le cours d'une élection. Certaines élections, comme les gouvernements étudiants, les clubs et les groupes en ligne, sont généralement considérées comme des « enjeux faibles », alors que les élections fédérales ou municipales sont des « enjeux élevés ». Ceci est cohérent avec la littérature de recherche sur le sujet (voir, par exemple [10]).

Adversaire	Capacité de l'attaquant				
	Supprimer le bulletin de vote	Apprendre le vote secret	Modifier le bulletin de vote	Apprendre l'identité de l'utilisateur	Apprendre l'adresse IP de l'utilisateur
Réseau passif (§5.3)		✓			✓
Réseau actif (§5.3) Service	✓	✓			✓
d'ID tiers. (§5.4)	✓			✓	✓
Racine sur l'appareil (§5.1)	✓	✓	✓	✓	✓
Serveur API Voatz (§5.2)	✓	✓	✓	✓	✓

Tableau 1 : Résumé des attaques potentielles par type d'adversaire : Ici, nous montrons quel type d'adversaire est capable d'exécuter quel type d'attaque ; par exemple, un adversaire du réseau passif est capable d'apprendre le vote secret d'un utilisateur et l'adresse IP de l'utilisateur. La viabilité de ces attaques peut dépendre de la configuration de l'élection particulière (le style de scrutin, les métadonnées, etc.), voir la section pertinente répertoriée pour des détails explicites.

à l'ingérence de la Russie dans l'élection présidentielle américaine de 2016. Des projets de loi ont été présentés au Sénat américain [48] et à la Chambre [70] qui visent à fournir des fonds pour réorganiser l'infrastructure électorale. Dans le même temps, il y a eu un regain d'intérêt pour la cryptographie en raison des progrès récents des systèmes responsables et transparents tels que la blockchain [57], et de la prolifération des appareils mobiles transportant des enclaves sécurisées basées sur du matériel pour les opérations cryptographiques ainsi que la biométrie.

Le résultat est une spéculation accrue sur la façon dont les appareils mobiles peuvent être utilisés pour permettre en toute sécurité de voter sur Internet. Au moment d'écrire ces lignes, au moins quatre entreprises tentent d'offrir des solutions de vote par Internet ou mobile pour les élections à enjeux élevés [56], et un candidat démocrate à la présidentielle de 2020 a inclus le vote à partir d'un appareil mobile via la blockchain dans sa politique [11]. A notre connaissance, seul Voatz a mis en service avec succès un tel système.

Malheureusement, les informations publiques sur le système de Voatz sont incomplètes. La FAQ [6], le blog et le livre blanc de Voatz [50] ne fournissent qu'une vague description de leur système global et de leur modèle de menace ; Voatz affirme qu'il exploite une combinaison d' une blockchain autorisée, de la biométrie et de magasins de clés soutenus par du matériel pour fournir des bulletins de vote chiffrés de bout en bout et vérifiables par les électeurs . Cependant, malgré les appels à publier une analyse plus détaillée et les préoccupations soulevées par de nombreux membres de la communauté de la sécurité électorale [29, 60], ainsi que des représentants élus [63], Voatz a refusé de fournir des détails officiels, invoquant la nécessité de protéger leur propriété intellectuelle. propriété [71]. Pire encore, lorsqu'un chercheur de l'Université du Michigan a effectué une analyse dynamique de l'application Voatz en 2018, l'entreprise a traité le chercheur comme un acteur malveillant et a signalé l'incident aux autorités. Cela a conduit le FBI à mener une enquête contre le chercheur [44, 47, 51, 75].

Cette position opaque est une menace pour l'intégrité du processus électoral. Compte tenu de la nature controversée des élections à enjeux élevés, des exigences de sécurité strictes des systèmes de vote et de la possibilité d'ingérence future d' agences de renseignement gouvernementales étrangères, il est crucial que les détails de tout système électoral sur le terrain soient analysables par le public. Dans toute démocratie, la légitimité du gouvernement repose sur le contrôle et la transparence du processus démocratique pour garantir qu'aucune partie ou acteur extérieur ne puisse modifier indûment le résultat.

D'un point de vue méthodologique, notre analyse a été considérablement compliquée par le manque de transparence de Voatz — à notre connaissance, lors d'examens de sécurité antérieurs de systèmes de vote par Internet déployés (voir Suisse [42], Moscou [37], Estonie [68] et Washington DC [74]), les chercheurs ont bénéficié d'informations importantes sur l'infrastructure de vote, y compris souvent la conception du système et le code source du système lui-même. .

Nous avons plutôt été contraints d'adopter une approche purement boîte noire et d'effectuer notre analyse sur une réimplémentation en salle blanche du serveur obtenue par rétro-ingénierie de l'application Android accessible au public de Voatz. Nous montrons que, malgré l'effort accru et les risques de validité, notre analyse est suffisante pour bien comprendre les défauts de Voatz . En particulier, nous démontrons que nos attaques résistent aux hypothèses optimistes pour les parties inconnues de l'infrastructure de Voatz (voir §5).

Le reste de l'article est organisé comme suit : Nous commençons au §2 par un bref historique des exigences de sécurité des élections , les revendications de sécurité de Voatz et des travaux connus analysant Voatz. Nous continuons au §3 en décrivant notre méthodologie d'ingénierie inverse, et discutons de la façon dont nous minimisons les menaces à la validité. Au §4, nous illustrons le système de Voatz tel que découvert dans notre méthodologie, y compris toutes les parties du processus de vote, l' infrastructure du serveur, la cryptographie personnalisée utilisée, et fournissons une brève discussion des facteurs que nous n'avons pas pu confirmer dans notre analyse. Ensuite, le §5 énumère les attaques découvertes dans notre analyse de Voatz. Nous concluons par une discussion au §6 pour fournir des leçons apprises et des recommandations pour les décideurs politiques dans cet espace pour aller de l'avant.

2. Arrière plan

Dans cette section, nous décrivons certaines des exigences de sécurité couramment observées dans les systèmes de vote cryptographiques proposés. Nous discutons ensuite des affirmations de Voatz et concluons en donnant un aperçu des analyses antérieures de Voatz.

Voter comme sujet de recherche à la fois dans la découverte appliquée de vulnérabilités et dans la cryptographie n'est pas nouveau. Vous trouverez ci-dessous une brève description des définitions de sécurité couramment utilisées dans le vote.

littérature du système.

Exactitude et facilité d'utilisation : pour garantir la légitimité de l' élection, un système de vote doit montrer de manière convaincante que tous les votes éligibles ont été exprimés comme prévu, collectés comme exprimés et comptés comme collectés [19].

Liberté de réception, confidentialité et résistance à la coercition : les systèmes de vote à scrutin secret doivent garantir que 1) aucun électeur ne peut prouver ses sélections (absence de réception), 2) qu'aucun choix d'électeur ne peut être subrepticement publié ou déduit (confidentialité), et 3) qu'un électeur ne peut pas coopérer avec un coerciteur pour prouver la façon dont il a voté (résistance à la coercition). Ces propriétés sont nécessaires pour assurer une élection libre de toute influence indue : si un électeur est en mesure de prouver la manière dont il a voté, il peut vendre son vote, et si les préférences d'un électeur sont divulguées ou forcées à être révélées, il peut être harcelé et contraint. cion [20, 30].

Vérification de bout en bout : les systèmes de vote vérifiables de bout en bout (E2E-V) ont la propriété que les électeurs reçoivent la preuve que leurs sélections ont été incluses, sans modification, dans le décompte final de tous les bulletins de vote recueillis, sans avoir besoin de faire confiance toute autorité distincte pour le faire. Des prototypes de recherche ont été développés qui offrent de telles garanties tout en maintenant la résistance à la coercition, la confidentialité et la liberté de réception en utilisant des techniques telles que la cryptographie visuelle, la cryptographie homomorphe , l'encre invisible et les réseaux mixtes [17, 23, 25, 65].

2.1 Demandes de garantie de Voatz

Bien qu'il n'y ait pas de description publique et formelle de leur système, Voatz fait un certain nombre d'affirmations sur les propriétés de sécurité de leur système via leur FAQ [6].

Immuabilité via une blockchain autorisée : Voatz affirme qu'une fois qu'un vote a été soumis, Voatz utilise "... la technologie de la blockchain pour s'assurer que... les votes sont vérifiés et stockés de manière immuable sur plusieurs serveurs de vérification géographiquement diversifiés". La FAQ va plus en détail, discutant de la fourniture de jetons pour chaque mesure de vote et candidat.

Cryptage des votes de bout en bout : Voatz fait plusieurs références aux votes eux-mêmes cryptés « de bout en bout ». À la connaissance des auteurs, il n'existe pas de définition formelle du « chiffrement des votes de bout en bout » ; par exemple, on ne sait pas où se trouvent les « extrémités » d'un système de vote chiffré de bout en bout. Il convient de noter qu'il existe des schémas de cryptographie homomorphe qui comptabilisent les votes sur les textes chiffrés des votes, de sorte qu'il suffit de déchiffrer un vote agrégé, en préservant la confidentialité des électeurs individuels [18], mais la FAQ ne précise pas si c'est ce que Voatz a l'intention.

Anonymat de l'électeur : Voatz affirme que "l'identité de l'électeur est doublement anonymisée" par le smartphone et la blockchain, et que "une fois soumises, toutes les informations sont anonymisées, acheminées via un 'mixnet' et publiées sur la blockchain".

Détection de la compromission de l'appareil : Voatz prétend utiliser plusieurs méthodes pour détecter si un appareil a été jailbreaké ou contient des logiciels malveillants, et que "l'application Voatz ne permet pas à un électeur de voter si le système d'exploitation a été compromis".

Piste d'audit vérifiée par l'électeur : Voatz affirme que les électeurs reçoivent un reçu numérique signé cryptographiquement de leur bulletin de vote après que leur vote a été soumis. Les garanties d'un tel reçu ne sont pas claires, bien que cela soit peut-être destiné à fournir des garanties similaires à celles des cryptosystèmes E2E-V.

2.2 Examen préalable de Voatz

Bien que nous soyons les premiers à publier une analyse approfondie de Voatz, d'autres ont exprimé des inquiétudes concernant leur système, les revendications de sécurité et le manque de transparence. Jefferson et al [29] ont compilé une longue liste de questions sans réponse sur Voatz, y compris l'utilisation par l'application d'un tiers, Jumio, en tant que service de vérification d'identité. Plusieurs auteurs ont observé le traitement électoral et l'audit du pilote Voatz lors des élections municipales de Denver en 2019, et ont constaté que l'activité principale de l'audit consistait à comparer un fichier PDF généré par un serveur d'un bulletin de vote avec le bloc blockchain enregistrant le même [43, 69]. Kevin Beaumont a trouvé ce qui semblait être plusieurs informations d'identification liées au service Voatz sur un compte Github public [14], et que le serveur Web Voatz exécutait plusieurs services non corrigés [15]. Voatz a répondu en citant un rapport du vérificateur SSL Qualys comme preuve de la sécurité du site [55], et a affirmé plus tard que le serveur non sécurisé identifié par Beaumont était une « opération de pot de miel » intentionnellement non sécurisée [73]. À la suite de cet examen public, en novembre 2019, le sénateur américain Ron Wyden a demandé à la NSA et au DoD de réaliser un audit de Voatz [63].

3 Méthodologie expérimentale

Comme effectuer une analyse de sécurité sur un serveur électoral en cours d'exécution soulèverait un certain nombre de problèmes juridiques et éthiques inacceptables [62], nous avons plutôt choisi d'effectuer toutes nos analyses dans un environnement de «salle blanche», en nous connectant uniquement à nos propres serveurs. Une attention particulière a été portée pour s'assurer que nos techniques d'analyse statique et dynamique ne pourraient jamais interférer avec Voatz ou tout autre service connexe, et nous avons fait de gros efforts pour que rien ne soit intentionnellement transmis aux serveurs de Voatz.2 Pour mieux comprendre l'infrastructure de Voatz, nous a commencé par décompiler la version la plus récente de son application Android3 telle qu'elle se trouve sur le Google Play Store au 1er janvier 20204 et a réimplémenté de manière itérative un serveur minimal qui effectue les processus d'élection comme visible depuis l'application elle-même. Cela comprenait les interactions impliquées dans l'appareil

[2] En effet, au moment de l'analyse, les serveurs de Voatz semblaient être en panne lorsqu'ils étaient testés avec une application non modifiée sur un appareil pris en charge et à jour.

[3] Nous n'avons effectué aucune analyse et ne faisons aucune déclaration concernant l'application iOS de Voatz.

[4] Version 1.1.60, SHA256 191927a013f6aae094c86392db4ecca825866ae62c6178589c02932563d142c1

l'inscription, l'identification des électeurs et le vote. Nous avons utilisé deux appareils pour notre analyse et notre développement dynamiques : un Pixel 2 XL pris en charge par Voatz sous Android 9 et un Xiaomi Mi 4i non pris en charge par Voatz sous Lineage OS avec An droid 8, tous deux jailbreakés avec le framework Magisk [2].

Afin de rediriger le contrôle vers notre propre serveur, nous avons été obligés d'apporter quelques modifications mineures au flux de contrôle de l'application. Pour réduire les menaces à la validité, nous avons limité ces modifications au minimum nécessaire afin de rediriger toutes les communications réseau. Nous:

1. Désactivation de l'épinglage du certificat et remplacement de tous les connexions à nos propres serveurs ;

2. Désactivation de la détection intégrée de logiciels malveillants et de jailbreak de l'application . Les détails sont disponibles au §5.1.1 ; et,

3. Suppression du cryptage supplémentaire entre l'appareil et tous les tiers encore actifs, reciblage de toutes les communications de ces services vers notre propre serveur et réimplémentation des parties nécessaires de leurs protocoles également.

Bien que tout cela aurait pu être accompli en modifiant statiquement le code du programme, nous avons plutôt opté pour modifier dynamiquement ou "accrocher" les parties pertinentes du code lors de l'exécution en utilisant un cadre de modding Android. Les modifications n'ont donc nécessité aucune modification du code de l'application elle-même, uniquement du code exécuté sur nos appareils de test, permettant un développement rapide et une transparence sur ce qui a été modifié à chaque étape de notre analyse.

Malgré cette longue description, notre base de code est relativement simple. Le code d'accrochage sur l'appareil se compose d'environ 500 lignes de Java qui exploite le Xposed Framework, une série de bibliothèques d'accrochage qui sont bien prises en charge et populaires dans la communauté de modding Android. Notre implémentation de serveur est d'environ 1200 lignes de code écrites en Python à l'aide du framework Web Flask.

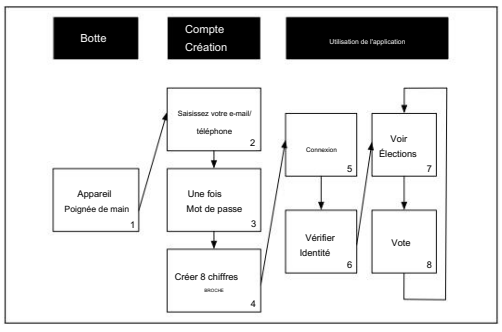


Figure 1 : Flux de travail de Voatz vu depuis l'appareil.

4 Conception du système de Voatz

Dans cette section, nous présentons l'infrastructure de Voatz telle que récupérée grâce à la méthodologie présentée au §3. Nous commençons par

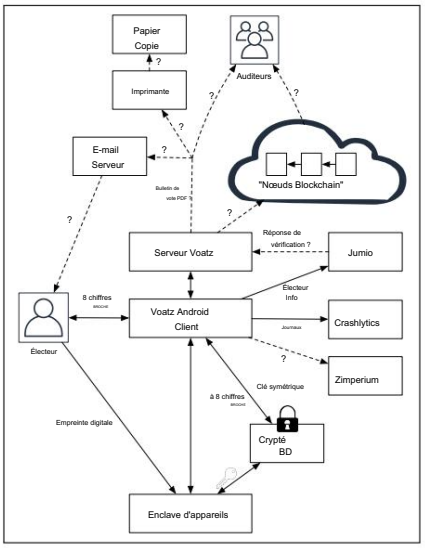


Figure 2 : Flux de données entre les composants Voatz et les services externes . On pense que des lignes pointillées existent mais n'ont pas été directement observées.

un aperçu du système §4.1, illustrant le processus par lequel l'appareil d'un utilisateur interagit avec l'application à toutes les étapes du processus de vote, y compris le protocole cryptographique personnalisé de Voatz §4.1.1, l'enregistrement de l'utilisateur et la vérification des électeurs §4.2, et le vote §4.3 . Enfin, nous discutons de toutes les mesures défensives côté appareil non protocolaires que nous avons découvertes §4.4.

4.1 Présentation du processus

La figure 1 présente un diagramme des étapes qui se produisent dans l'application , de la connexion au vote électoral. Ils sont:

1. L'appareil initie une poignée de main avec le serveur, créant une clé partagée qui permet une couche supplémentaire de chiffrement au-delà de TLS (encadré 1). La communication entre l'appareil et le serveur Voatz est décrite au §4.1.1.

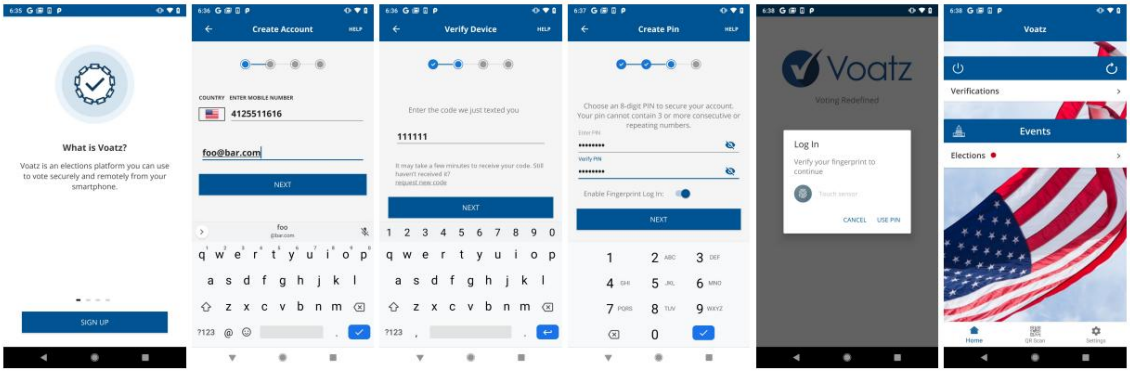
2. L'utilisateur crée un compte en fournissant son adresse e-mail, son numéro de téléphone et un code PIN à 8 chiffres (cases 2 à 4).

3. L'utilisateur se connecte avec ce code PIN (encadré 5).

4. L'utilisateur vérifie son identité, en utilisant l'intégration de Voatz avec un service tiers appelé Jumio (encadré 6). L'application demande une analyse de la photo d'identité de l'utilisateur, un enregistrement de son visage et de l'adresse de l'utilisateur, puis envoie toutes ces informations aux serveurs de Jumio.

5. L'utilisateur sélectionne dans une liste d'élections ouvertes, puis marque et soumet son bulletin de vote. Selon la configuration des élections, Voatz peut autoriser le « spoiling du vote 5 », donc

5L'annulation d'un vote fait référence à l'émission d'un nouveau vote qui invalide tous les bulletins de vote précédemment déposés.



(a) Écran initial. (b) Courriel et téléphone. (c) OTP. (d) Numéro d'identification. (e) Première connexion. (f) Écran principal.

Figure 3 : Le processus d'enregistrement de l'utilisateur, se connectant à notre réimplémentation de serveur.

ce processus peut être répété avant la clôture des élections. (Cases 7-8)

Communication La figure 2 montre la communication entre les composants de Voatz et d'autres entités. Comme nous n'avons pu observer directement que la communication impliquant l' application Voatz, le reste de ce diagramme est une tentative de reconstruction basée sur des documents publiés par Voatz [50] et par la Denver Elections Division [35].

Les trois principaux services tiers utilisés par l' application Voatz sont le service de vérification d'identité Jumio, un service de rapport d'incident Crashlytics et un service de sécurité des appareils Zim perium. Parmi ceux-ci, le plus important est Jumio, sur lequel Voatz s'appuie pour la vérification de l'identité, et à qui l'application envoie des informations personnelles substantielles (voir §4.2).

4.1.1 Prise de contact et protocole du serveur Voatz

Le serveur de Voatz est implémenté en tant qu'application REST - toutes les communications entre le serveur de Voatz et l'application se produisent sous la forme d'une série de commandes HTTPS GET, PUT et POST encodées en JSON. Le serveur REST de l'application est voatzapi.nimsim.com , voatz.com n'étant utilisé que pour les actifs statiques tels que les images et le texte. Toutes les parties du protocole tirent parti de la pile TLS intégrée du système d'exploitation Android et utilisent l'épinglage de certificats pour garantir que le certificat entrant provient d'une autorité de certification émettrice particulière.

Ensuite, en plus de TLS, le système effectue une « poignée de main de l'appareil » avec les étapes suivantes :

1. L'application génère 100 paires de clés ECDSA SECP256R1 et envoie au serveur les 100 clés publiques correspondantes. L'appareil n'enregistre que la 57e paire de clés (PKD, SKD).

2. Le serveur génère 100 paires de clés ECDSA SECP256R1, sélectionne la 57e (PKS,SKS) et effectue le reste d'un échange de clés ECDH pour générer un secret partagé (SKecdh).

3. Le serveur génère les paramètres AES-GCM ; une clé symétrique aléatoire AES-GCM 256 bits (SKaes), un nonce aléatoire 16 bits (N) et une étiquette (T).

4. Le serveur envoie ensuite à l'appareil les 100 clés publiques générées ci-dessus, y compris le PKS comme 57e clé et ECDH-Encrypt(SKecdh,SKaes||N||T)

5. Sur les 100 clés publiques envoyées par le serveur, l'application sélectionne la 57e clé publique (PKS) et termine la poignée de main ECDH pour créer la clé partagée ECDH SKecdh. Enfin, il décrypte et analyse les paramètres AES-GCM (SKaes, N, T).

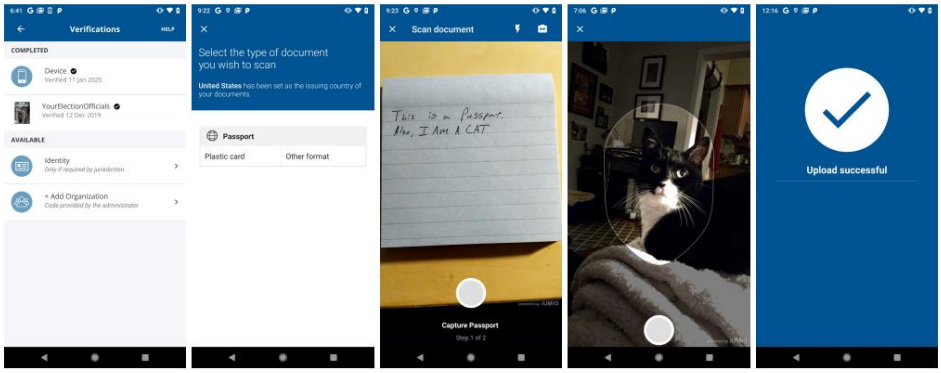
Cette poignée de main est effectuée chaque fois que l'application est lancée pour la première fois et, à partir de ce stade de l'exécution de l'application, chaque communication entre l'application et le serveur est cryptée à l'aide de l'algorithme standard AES-GCM via SKaes, en plus au cryptage fourni par TLS. Notez qu'il n'y a pas d'authentification des clés ECDSA par l'application, au-delà des certificats TLS d'encapsulation. Cela a rendu très simple le reciblage du serveur - nous avons remplacé toutes les URL requises dans l'application par les nôtres et avons suivi le protocole.

De plus, cela rend l'utilisation de la poignée de main quelque peu floue, car elle n'offre aucune protection contre les attaques MITM actives par rapport à l'authentification déjà fournie par TLS.

Il convient également de mentionner que toutes les clés sauf la 57e sont immédiatement abandonnées du côté de l'appareil - à la fois les clés secrètes supplémentaires générées par l'appareil lors de la première étape et les clés publiques qu'il reçoit du serveur. Nous concluons que cet échange de 100 clés est probablement une tentative d'obscurcissement, plutôt que de servir un but utile au protocole de sécurité.

4.2 Enregistrement de l'utilisateur et vérification de l'identité

Une fois que l'application a terminé la poignée de main de l'appareil, l'utilisateur peut commencer le processus d'enregistrement, comme le montre la figure 3. Ici, l'utilisateur est invité à soumettre son e-mail et son numéro de téléphone, et à effectuer une opération de mot de passe à usage unique via



(a) Vérification fragment. (b) Document se lecteur. (c) Image d'un identifiant de (d) Visage "selfie". (e) Vérification telle cess.

Figure 4 : Le processus de vérification des électeurs vu de notre environnement expérimental.

SMS. Enfin, l'utilisateur sélectionne un code PIN à 8 chiffres qui est ensuite envoyé au serveur et largement utilisé pour l'authentification de l'utilisateur .

Si l'utilisateur a une empreinte digitale enregistrée sur son appareil, il a la possibilité « d'enregistrer » son empreinte digitale comme mécanisme d'authentification alternatif. En effet, cela fonctionne en stockant le code PIN sur le disque, crypté à l'aide d'une clé liée biométriquement à l'empreinte digitale de l'utilisateur via le Keystore Android.

Android Keystore est un service système qui, s'il est utilisé correctement, effectuera diverses opérations cryptographiques pour le compte de l'application, sur les données au niveau de l'application, sans exposer le matériel de clé requis à la mémoire hôte de l' application. matériel, ces clés au niveau de l'appareil sont stockées dans le matériel protégé du fabricant et peuvent être conçues pour demander à l'utilisateur d'entrer le mot de passe ou l'empreinte digitale de son appareil avant de les utiliser.

Après l'enregistrement, l'utilisateur est invité à se connecter via le code PIN (ou le décryptage des empreintes digitales du code PIN). Outre le code PIN, quatre informations sont envoyées au serveur pour authentifier l'utilisateur lors de la connexion : un identifiant d'appareil unique généré via le système ANDROID_ID d'Android,7 un numéro d'identification client, une valeur "nextKey" et un " auditToken ». NextKey et auditToken sont initialement reçus du serveur d'API, ne sont jamais modifiés sauf lorsqu'ils sont mis à jour par le serveur et ne semblent pas être utilisés dans la cryptographie côté appareil. La façon dont ces paramètres d'authentification sont stockés est explorée au §4.4.

Après l'authentification, l'utilisateur peut encore avoir besoin de fournir une preuve d'identité, ce qui nécessite de visiter le menu de vérification à partir de l'écran principal (Figure 4a). Lorsque l'utilisateur sélectionne l'option d'identité, l'application lance la sous-activité de Jumio pour sélectionner un type de document (Figure 4b). L'utilisateur est invité à prendre une photo de sa carte d'identité ou de son passeport (4c) et à prendre une photo de selfie (4d), après quoi une boîte de dialogue invite l'utilisateur à entrer son adresse de vote enregistrée (non illustrée). L'application télécharge ensuite

6Voir la documentation Keystore d'Android pour plus de détails [12].

7Voir [13] pour plus d'informations sur les UUID d'appareils locaux d'Android.

données au serveur de Jumio, y compris la photo de l'utilisateur, le nom, l'adresse et la photo d'identité de l'électeur (4e).8 Enfin, après avoir reçu une réponse du serveur de Jumio, l'application envoie également un sous-ensemble des données de l'utilisateur au serveur de Voatz.

Il convient de noter que le petit logo translucide dans le coin inférieur droit des photos prises au cours de ce processus (Figures 4c, 4d) semble être la seule indication intégrée à l'utilisateur que Jumio existe, et le seul moyen pour un utilisateur serait conscient que ces données sont envoyées à un tiers.

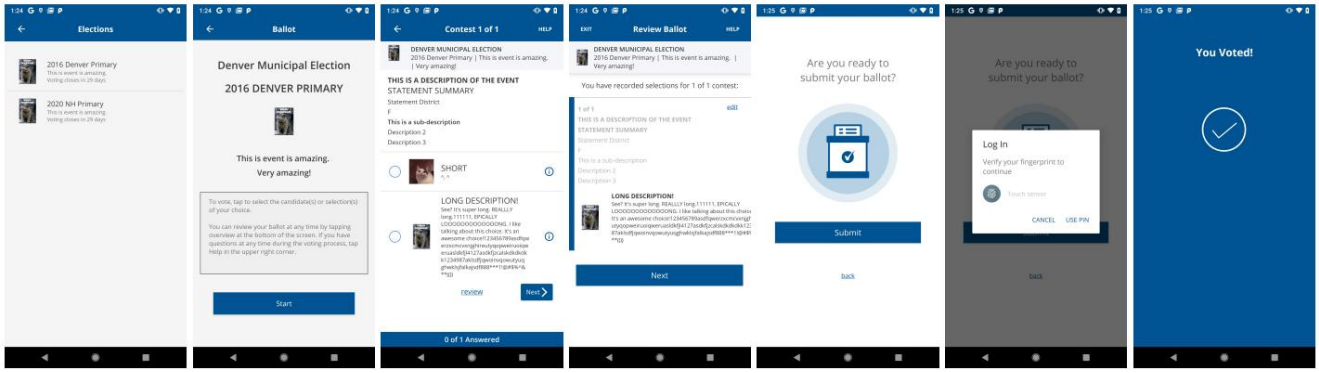
4.3 Vote

Une fois l'utilisateur vérifié, l'application interroge le serveur pour obtenir des données de configuration relatives aux événements auxquels l'électeur est autorisé à participer, activant un menu permettant à l'utilisateur de sélectionner les événements disponibles (voir Figure 5). Ces données de configuration comprennent tous les événements auxquels l'électeur a accès, les bulletins de vote de ces événements, les questions particulières de chaque bulletin de vote et les options disponibles pour ces questions.

Le votant commence par sélectionner un événement (5a), puis peut visualiser les questions associées à ces événements particuliers, sélectionner des réponses (ou pas de réponse du tout, selon la configuration de l'événement) et soumettre sa réponse au serveur. Au moment de la soumission, l'utilisateur est à nouveau invité à déchiffrer son code PIN (5e), qui est utilisé comme mécanisme d'authentification final avant que le bulletin ne soit soumis au serveur.

Il est important de noter que le vote n'est pas soumis directement à un système de type blockchain, mais est plutôt soumis via ce serveur API. De plus, bien qu'il soit demandé à l'utilisateur de s'authentifier avant la soumission, au-delà du MAC associé à l'algorithme AES-GCM et englobant la session TLS, le texte du vote lui-même n'est pas autrement signé. La seule indication de jetons de type blockchain soumis ou échangés

8De plus, Jumio lui-même a révélé qu'il utilise un tiers, Facetec, pour l'aider à analyser les selfies vidéo [7]. Comme nous n'avons pas de visibilité sur leurs services, nous ne pouvons pas confirmer si Jumio transmet effectivement ou non des informations aux serveurs contrôlés par Facetec.



(a) Sélection de l'événement. (b) Vote. (c) Questions. (d) Examen. (e) Soumission. (f) Déchiffrement du code PIN. (g) Succès.

Figure 5 : Le processus de vote tel qu'il apparaît dans une élection fictive que nous avons créée pour cette expérience.

est le "auditToken", mais cette chaîne n'est jamais modifiée par l'application et semble être une valeur statique unique. La figure 10 montre l' intégralité de ce qui est envoyé au serveur, crypté AES-GCM, après qu'un utilisateur a soumis son vote.

4.4 Mesures défensives côté appareil

Au cours de l'exécution de notre analyse, nous avons découvert que Voatz utilise un certain nombre de techniques d'obscurcissement, exploite un service d'analyse de virus tiers et utilise une base de données cryptée sur l'appareil pour protéger les données sensibles stockées localement.

Base de données cryptée sur disque : une fois l'inscription terminée, les identifiants de connexion de l'utilisateur (NextKey, auditToken et numéro d'identification client), ainsi que l' historique complet des votes de l'électeur, sont stockés localement dans une base de données cryptée à l'aide de la structure de base de données Realm. [4]. Lorsque l'application de Voatz tente d'interroger la base de données, le Keystore demande à l'utilisateur de s'authentifier via une empreinte digitale ou un code PIN (voir Figure 5f), avant d'effectuer les opérations requises.

La clé de la base de données est directement liée au PIN de l'utilisateur ; plus précisément, le système exécute PBKDF2 avec SHA1 sur le code PIN pour générer la clé. Rappelez-vous que cela permet au système d'utiliser une empreinte digitale comme méthode alternative de déchiffrement de la base de données - Lors de la connexion, l'application peut s'authentifier via l'empreinte digitale pour déchiffrer le code PIN, ou utiliser le code PIN directement pour déchiffrer la base de données et accéder au reste de l'application.

Détection de logiciels malveillants tiers (Zimperium) : Voatz exploite une solution antivirus tierce appelée Zimperium.
Au moment de l'initialisation, l'application Voatz charge le code de Zimperium en tant que service distinct et enregistre une série de rappels qui alerteront le serveur API si Zimperium détecte une menace. Ce message inclut les détails de la menace, l'ID utilisateur et l'ID de l'appareil, ainsi que l'adresse IP de l'appareil incriminé.

Les analyses de Zimperium incluent (mais ne sont pas limitées à) des preuves de concept d'exploit connus, des logiciels malveillants connus et des indicateurs que l'utilisateur a installé des outils de superutilisateur connus indiquant un appareil rooté/jailbreaké. De plus, Zimperium déclenchera des rappels si l'utilisateur semble avoir activé les fonctionnalités de débogage local d'Android telles que le débogage adb à distance.

Obfuscation partielle du code et compression : sans que le développeur ne prenne des précautions supplémentaires, les applications Android peuvent être décompressées en lecture et décompilées à proximité de la source d'origine via des outils faciles à utiliser tels que APKTool [1] et JADX [66]. Cependant , une grande partie de l'application Voatz est masquée à l'aide d'un packer qui présente plusieurs obstacles à l'analyse.

Tout d'abord, de nombreuses classes et noms de fonctions ont été renommés en chaînes Unicode aléatoires. En plus de rendre la décompilation résultante plus difficile à lire, cette obfuscation a également provoqué le plantage d'APKTool, tandis que JADX a terminé avec succès la décompilation, mais a laissé de nombreux fichiers de ressources (y compris les chaînes d'application et les images) illisibles. L'application de Voatz contenait également quelques fichiers zip qui semblent effectuer une attaque à la bombe zip [34], qui va à l'encontre de certaines implémentations de décompression. Enfin, toutes les bibliothèques natives tierces incluses pour ARM n'ont pas pu s'ouvrir dans notre version d'IDA, bien qu'il ne soit pas clair s'il s'agissait d'une mesure défensive active car elles ont été désassemblées avec succès à l'aide de Ghidra.

Nous avons pu vaincre l'obscurcissement grâce à une analyse manuelle intensive et, dans certains cas, nous avons été aidés à récupérer les noms de variables d'origine par l'application elle-même. Premièrement, l'application utilise de nombreuses bibliothèques qui dépendent en interne de la réflexion Java, ce qui rend l'obfuscateur incapable de renommer les classes ou les méthodes référencées de cette manière. Deuxièmement, l'application et certaines de ses bibliothèques sont écrites en Kotlin. Bien que certains idiomes Kotlin ne se décompilent pas facilement en Java, l'utilisation globale de Kotlin facilite l'ingénierie inverse - le compilateur Kotlin insère de nombreux contrôles d'exécution dans le code, chacun comprenant une chaîne avec un message d'erreur à afficher en cas d'échec. Les noms de classe, de fonction et de variable sont souvent stockés dans ces chaînes.

Obfuscation des chaînes Pour compliquer davantage l'analyse statique, les chaînes qui contrôlent les paramètres cryptographiques de la poignée de main de l'appareil (par exemple « AES-GCM ») sont obscurcies avec un schéma basé sur XOR, puis automatiquement désobscurcies lors de l'exécution. Comme les chaînes masquées de cette manière incluent les messages d'erreur générés par le compilateur Kotlin, cela semble être le résultat d'un outil automatisé qui n'a été activé que pour ces méthodes particulières.

4.5 Parties non confirmées du processus

Comme nous n'avons pas accès aux serveurs de Voatz et avons délibérément évité toute interaction avec eux, il existe malheureusement quelques cas où nous ne sommes pas en mesure de confirmer le comportement de certains acteurs tiers du système.

Confirmation d'exécution de Zimperium : Zimperium peut communiquer à ses propres serveurs pour confirmer que le service est en cours d'exécution, puis communiquer si Zimperium est actif directement à Voatz. À notre connaissance, il n'existe aucune documentation publique suggérant que c'est ainsi que fonctionne Zimperium, et nous ne trouvons aucune indication dans les rappels associés à Zimperium que cela se produise, voir §5.1.1.

Confirmation de l'électeur Jumio : la documentation de Jumio traite en détail de la possibilité facultative de communiquer avec les serveurs de Jumio pour la vérification hors bande d'un utilisateur. Comme il s'agit d'une fonctionnalité bien documentée du système, nous supposons que le serveur API de Voatz reçoit une confirmation directement des serveurs de Jumio pour la vérification de l'identité.

Reçus de bulletins de vote et blockchain : selon un livre blanc de Voatz, les votes sont enregistrés sur une blockchain autorisée à 32 nœuds répartis sur plusieurs centres de données Amazon AWS et Mi crosoft Azure [35]. Des images de l'audit des élections municipales de Denver de 2019 montrent que le processus d'audit consiste à inspecter manuellement les blocs de blockchain indiquant les transactions, en obtenant plusieurs champs, y compris un hachage des choix de l'électeur. L'auditeur compare ensuite manuellement le hachage via une table de recherche à un PDF affichant les choix de l'électeur. Ces fichiers PDF seraient également imprimés par l'autorité électorale sous forme de document papier et seraient des versions expurgées du reçu envoyé par courrier électronique aux électeurs. Bien que nous sachions que, lors de l'élection de Denver, de nombreux électeurs ont répondu manuellement pour indiquer qu'ils avaient reçu un reçu, rien ne prouve que Voatz puisse vérifier automatiquement la livraison du reçu [43].

Dans notre exploration du code, nous ne trouvons aucune indication que l'application reçoit ou valide un enregistrement qui a été authentifié ou stocké dans une quelconque forme de blockchain. Nous n'avons en outre trouvé aucune référence à des chaînes de hachage, des journaux de transparence ou d'autres preuves cryptographiques d'inclusion. Nous concluons que toute utilisation d'une blockchain par Voatz a probablement lieu uniquement sur le backend, ou au stade de la réception via l'utilisation d'un autre mécanisme.

Les seules références aux reçus d'électeur dans l'application proviennent d'une boîte de dialogue qui demande un mot de passe au serveur et d'un lecteur de code QR (apparemment non implémenté). Le texte de la boîte de dialogue de reçu d'électeur semble confirmer que les reçus de vote sont bien envoyés à l'électeur par e-mail et chiffrés avec le mot de passe fourni par le serveur (voir Figure 6). Le lecteur de code QR de Voatz a un code fonctionnel pour une méthode hors bande de réception des identifiants d'organisation, qui permet à l'électeur de participer à des événements particuliers, et un talon largement non implémenté pour vérifier un vote - tenter de scanner un code QR qui démarrer le processus de vérification du vote entraînera le message « pas encore pris en charge » présenté à la figure 6.

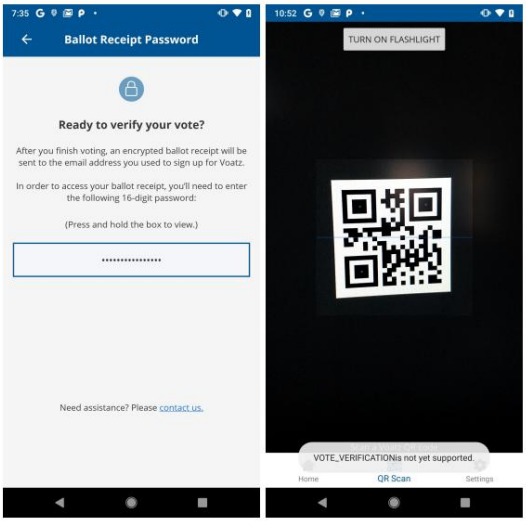


Figure 6 : A gauche : l'écran de demande de mot de passe. A droite : l'écran de capture du code QR ; notez la fenêtre contextuelle indiquant que le type de code QR VOTE_VERIFICATION n'est pas implémenté.

5 Analyse et attaques

Dans cette section, nous explorons diverses attaques en assumant le rôle d'un adversaire qui contrôle certaines parties du système électoral. Cela comprend trois adversaires avec différents niveaux d'accès à des parties individuelles de l'infrastructure globale :

1. Un attaquant qui contrôle l'appareil d'un utilisateur,
2. Un attaquant qui contrôle le serveur API de Voatz, et
3. Un adversaire du réseau qui peut intercepter l'activité du réseau entre l'appareil de l'électeur et le serveur API, mais n'a plus accès.

Nous pensons que ces adversaires sont crédibles compte tenu de la nature des enjeux élevés des élections dans lesquelles Voatz est destiné à être utilisé, et des ressources des attaquants associés. Obtenir le contrôle root de l'appareil d'un utilisateur peut se faire par un certain nombre de moyens nécessitant différents niveaux de compétence - via un logiciel malveillant, un

partenaire intime ou conjoint, dans le cadre d'un passage frontalier, etc. Les adversaires du réseau peuvent se présenter sous de nombreuses formes similaires, y compris ceux qui exploitent le routeur domestique d'un utilisateur (qui sont notoirement peu sûrs [39, 40]), le wifi non crypté d'un café à partir duquel un utilisateur tente de voter, ou le FAI de l'utilisateur.

Inclure le serveur API de Voatz dans cette analyse est utile pour un certain nombre de raisons. Bien que l'accès au serveur de Voatz puisse être plus difficile que l'appareil de l'utilisateur et/ou l'infrastructure réseau entre le serveur et l'utilisateur, si l'utilisation de Voatz devait être augmentée au point que leur base d'utilisateurs puisse modifier le résultat d'une élection, il est il n'est pas impossible pour eux d'être la cible des États-nations, auquel cas il n'est pas non plus hors de question que les agences de renseignement dépensent des ressources considérables, tirant parti des vulnérabilités non divulguées du jour 0, de l'espionnage, de la coercition ou des attaques physiques, pour accéder à des systèmes cruciaux ou à du matériel clé. De plus, une promesse clé de la blockchain est qu'elle fournit un environnement dans lequel l'électeur et l'autorité électorale peuvent faire confiance au système, plutôt qu'à Voatz, que l'élection s'est déroulée correctement.

Hypothèses et menaces à la validité Comme nous manquons de détails concrets sur la mise en œuvre de l'infrastructure du serveur ou du back- end, nous ne pouvons pas faire d'hypothèses sur ce que Voatz enregistre sur sa blockchain, la sécurité opérationnelle de ses serveurs, sa blockchain ou les clés cryptographiques utilisées.

Pour limiter les risques de validité, notre analyse ne fera aucune hypothèse sur l'état du serveur au-delà de ce que nous pouvons glaner de l'application elle-même, et nous supposerons que toutes les interactions, y compris toutes les activités cryptographiques telles que vues depuis l'appareil au §4.1 .1, sont enregistrés dans la blockchain et que ces enregistrements de blockchain sont sécurisés, surveillés et immuables. Cela inclut tous les textes chiffrés du protocole, ainsi que tout caractère aléatoire utilisé dans les algorithmes.

Notez qu'il s'agit d'une analyse optimiste de l'utilisation de la blockchain dans ce système. Il est peu probable que chaque interaction soit stockée via la blockchain, et la documentation de Voatz sur les élections en Virginie-Occidentale indique que les serveurs de vérification sont répartis également entre Amazon AWS et Azure de Microsoft , ce qui indique que leur schéma est vulnérable à Microsoft ou Amazon ajoutant subrepticement des ressources et exécutant une attaque à 51 %, ou effectuer une attaque de minage égoïste qui ne nécessite que 1/3 de la puissance de calcul [32].

Néanmoins, nous nous concentrons sur ce qui est prouvable compte tenu de notre accès limité au système, et montrons que cette analyse est suffisante pour démontrer un certain nombre d'attaques significatives.

5.1 Attaques côté client

Nous constatons qu'un attaquant avec des privilèges root sur l'appareil peut désactiver les protections basées sur l'hôte de Voatz, et donc contrôler furtivement le vote de l'utilisateur, exposer son vote privé et exfiltrer le code PIN de l'utilisateur et d'autres données utilisées pour s'authentifier auprès du serveur

```
argClass = loadClass("com.zimperium.DetectionCallback");

findAndHookMethod("com.zimperium.ZDetection", chargeur,
   "addDetectionCallback", argClass, nouveau XC_MethodHook() {
      void beforeHookedMethod(MethodHookParam p) {
         p.setResult(null); // empêche l'exécution de la méthode
      }
});
```

Figure 7 : Code simplifié pour désactiver le SDK de sécurité Zimperium.

5.1.1 Déjouer la détection des logiciels malveillants basée sur l'hôte

Le SDK Zimperium inclus dans Voatz est configuré pour détecter le débogage et d'autres tentatives de modification de l'application, et pour collecter des informations sur tout logiciel malveillant qu'il trouve. Par défaut, il aurait détecté notre analyse de sécurité, empêché l'application de fonctionner normalement et alerté le serveur API de nos actions.

Comme mentionné au §4.4, Zimperium communique avec l' application Voatz, et finalement avec le serveur API de Voatz, via un ensemble de rappels initiés lors du chargement de l'application. Vaincre Zimperium était donc aussi simple que de remplacer ses points d'entrée pour empêcher le SDK de s'exécuter. Les utilitaires d'accrochage fournis par Xposed Framework nous permettent de détourner le flux de contrôle avec un minimum d'effort — la figure 7 montre le code pour désactiver l'un de ses deux points d'entrée ; au total, la désactivation de Zimperium a nécessité quatre lignes de code, et est imperceptible pour l'utilisateur.

Nous supposons qu'il n'y a pas de communication hors bande entre Zimperium et Voatz, et ne trouvons aucune indication dans la documentation de Zimperium ou dans notre analyse de l'application que ce service existe. Si une telle communication existe, elle n'augmenterait que marginalement l'effort nécessaire pour la vaincre ; il faudrait accrocher d'autres parties de Zimperium qui effectuent la détection, ou communiquer directement avec leur serveur.

5.1.2 Contrôle total sur l'utilisateur, sur ou hors appareil

Une fois la détection de logiciels malveillants basée sur l'hôte neutralisée, un attaquant disposant de privilèges root a la capacité de contrôler complètement les actions de l'utilisateur et la vue de l'application, ainsi que de divulguer les décisions de vote et les informations personnelles de l'utilisateur.

Vol des données d'authentification de l'utilisateur : bien qu'ils soient cryptés avec des clés qui exploitent le Keystore Android, le code PIN de l'utilisateur et les autres informations de connexion ne sont pas stockés dans un stockage protégé et passent par la mémoire de l'application. L'exfiltration de ces informations clés permettrait à un attaquant distant de se faire passer pour l'utilisateur directement sur les serveurs de Voatz, même hors de l'appareil.

Nous constatons qu'un attaquant disposant d'un accès root à l'appareil peut subrepticement voler le code PIN et le reste des données d'authentification de Voatz. Au cours de l'exécution de notre analyse, nous avons développé un outil qui intercepte et enregistre toutes les communications entre l'appareil et le serveur avant qu'elles ne soient cryptées avec SKaes, ainsi qu'avant que les données ne soient cryptées et stockées dans le

base de données locale. Cela nous a permis de voir, en clair, à la fois le code PIN brut de l'utilisateur et d'autres données d'authentification. Bien que notre preuve de concept s'arrête à enregistrer ces informations via les fonctionnalités de débogage du système d'Android (adb logcat), il serait trivial de diffuser ces requêtes sur le réseau, de les modifier ou de les empêcher de se produire.

Un attaquant n'a pas nécessairement besoin d'attendre que l'utilisateur décide de voter - des attaques hors ligne contre le stratagème de Voatz sont également tout à fait possibles. Rappelez-vous que la base de données ne nécessite que le code PIN de l'utilisateur pour se déverrouiller et ne limite en aucun cas le nombre de tentatives de ce code PIN. Pire, l'application limite artificiellement le code PIN à exactement 8 caractères numériques, ce qui signifie qu'il n'y a que 100 000 000 codes PIN possibles.9 Une attaque par force brute peut donc facilement redécouvrir le code PIN en générant à plusieurs reprises des clés et en tentant de déchiffrer la base de données, en récupérant le code PIN, en se connectant informations et l'historique des votes de

l'utilisateur en même temps.10 Une telle attaque par force brute peut être effectuée assez rapidement. Notez qu'un attaquant n'a pas besoin de le faire sur l'appareil, car le fichier de base de données crypté peut être exporté. Nous avons implémenté un prototype de cette attaque et confirmé qu'un attaquant peut forcer brutalement la clé en environ deux jours sur un MacBook Pro 2017 à 3,1 GHz. Nous concluons qu'une telle menace est viable, en particulier si la même installation de Voatz sera utilisée lors de plusieurs élections.

Attaque furtive de modification de l'interface utilisateur : il est simple de modifier l'application afin qu'elle soumette tout vote souhaité par l'attaquant, tout en présentant la même interface utilisateur que si l'application enregistrait la soumission de l'utilisateur. Si la configuration des élections permet le vote-spoiling, il existe également une variante de cette attaque précédemment démontrée sur le système de vote électronique estonien : permettre à l'utilisateur de voter normalement , mais modifier le vote une fois que l'utilisateur ferme l'application [68].

De même, l'attaquant pourrait supprimer furtivement les choix de l'électeur s'il sélectionne un candidat indésirable, mais continuer à afficher la boîte de dialogue de vérification comme si le vote avait été émis avec succès. Pour l'autorité électorale, cela pourrait être indiscernable du fait que l'électeur n'a pas soumis de bulletin de vote. Pour l' électeur, cela est indiscernable du vote correct, du moins jusqu'à ce que l'autorité publie les registres des électeurs pour cette élection.11

5.2 Attaques de serveur

Nous constatons que, en supposant l'utilisation optimiste de la blockchain discutée dans le modèle de menace, le serveur de Voatz est toujours capable de violer subrepticement la vie privée de l'utilisateur, de modifier le vote de l'utilisateur et de contrôler le résultat de l'élection.

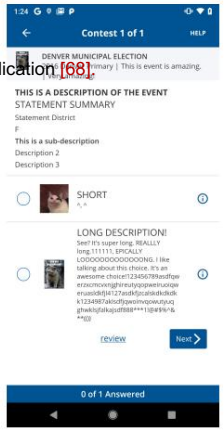
En particulier, nous constatons que le protocole discuté au §4.1.1 ne fournit aucune garantie contre le serveur API qui modifie activement utiliser, visualiser ou inventer une communication à partir de l'appareil ; le serveur peut exécuter une attaque MITM active entre l' appareil utilisateur et tout mécanisme blockchain ou mixnet existant à l'autre extrémité. Notez qu'aucune autre opération cryptographique n'est effectuée entre l'appareil et le serveur à aucun moment autre que le cryptage AES, y compris toute sorte de signature cryptographique par l'appareil ou le magasin de clés de l'appareil.

Si le serveur effectue lui-même ces opérations cryptographiques - que SKaes est disponible pour le serveur - il peut déchiffrer le bulletin de vote de l'utilisateur avant qu'il ne soit soumis à un journal externe et rechiffrer de manière convaincante toute valeur à envoyer au journal.

Même si SKaes n'est pas disponible pour le serveur - par exemple, si toutes les opérations cryptographiques sont effectuées dans un module de sécurité matériel (HSM) - il doit alors au moins avoir accès au flux TLS non chiffré, et il est donc toujours possible pour le serveur pour exécuter une attaque MITM active.

Rappelez-vous qu'il n'y a pas d'authentification par clé publique effectuée dans le cadre de la poignée de main de l'appareil, et il n'y a aucune preuve ou vérification par l'appareil que ces interactions sont jamais enregistrées sur la blockchain. Le serveur peut donc mettre fin à la connexion avant le HSM et usurper arbitrairement l'identité de l' appareil de l'utilisateur, par exemple en rejouant l'ensemble de la poignée de main de l'appareil et toutes les communications futures via le HSM vers la blockchain.12 Notez que, compte tenu de ces attaques, il n'est pas clair s'il y a existe un schéma dans lequel un reçu peut prouver de manière convaincante que le vote correct a été enregistré.

5.3 Adversaire du réseau



(une question.

```
Choice = { "
    choiceDetails " : {"imageUrl":SHORT_IMG,
                "webUrl":SHORT_IMG},
    "choiceId" : "1", "
    description "      : "Court"   ,
    " description 1": "^" "     ,
    description 2": "^" "       ,
    isWriteIn "      : Faux ,
    " non sélectionnable " : Faux
}
Choice2 =
    { "choiceDetails "   :  { 'imageUrl' :
                LONG_IMG_URL, 'webUrl' :
                LONG_IMG_URL},
    "choiceId" : "2", "
    description "      : "Description longue !" " ,
    description 1": "Tu vois ? C'est super long.
                VRAIMENT long.111111", "
    description 2": "ÉPIQUEMENT
                LOOOOOOOOOOOONG..." "
    isWriteIn "      : Faux " ,
    non sélectionnable ": Faux
}
```

(b) JSON correspondant.

Figure 8 : Attaque par canal de vote expliquée.

9Voatz interdit également les codes PIN contenant 3 chiffres consécutifs identiques, ce qui élimine ~ 5% de ceux-ci.

10A salt est également nécessaire pour déverrouiller la base de données. Ceci est stocké sur disque, non crypté, dans le fichier de préférences partagées de l'application.

11Pour les élections américaines, les archives publiques répertorient souvent les électeurs qui ont participé.

12Peut-être que cet hypothétique HSM contient également les clés TLS nécessaires pour mettre fin à la connexion, et effectue toutes les opérations cryptographiques dans l'enclave. Cependant, toutes les communications sont cryptées avec SKaes, y compris celles qui nécessitent des requêtes contre les bases de données des utilisateurs, il n'est donc pas clair que ce soit le cas, mais, même ainsi, le serveur est capable d'effectuer un certain nombre d'attaques sur l'utilisateur. Voir §5.3.

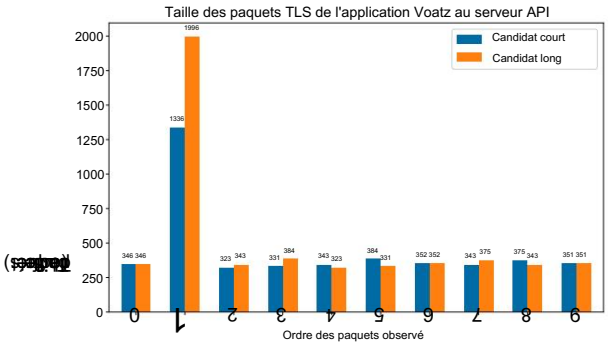


Figure 9 : Longueurs des paquets chiffrés TLS immédiatement après qu'un utilisateur a soumis un vote, dans l'ordre d'envoi. Notez la taille du candidat "court" et "long" dans le paquet 1.

Nous constatons qu'un adversaire ayant la possibilité de visualiser l' activité réseau de l'utilisateur, sans avoir accès à aucun élément clé, peut toujours déduire comment l'utilisateur a voté. Plus précisément, dans cette section, nous démontrons que l'application divulgue la longueur du texte brut, ce qui peut permettre à un attaquant de savoir, au minimum, pour quel candidat l'utilisateur a voté.

La vulnérabilité provient de la manière dont un bulletin de vote est soumis au serveur après qu'un utilisateur a fini de sélectionner ses options. Comme le montre la figure 10, la liste de « choix » dans une soumission de vote ne contient que les options sélectionnées par l'utilisateur, et inclut avec ce choix l'intégralité des métadonnées fournies par le serveur à propos de ce candidat. Ceci, à son tour, fait varier considérablement la longueur du texte chiffré en fonction des choix de l'électeur.

La figure 8b montre les différences de métadonnées envoyées vers et depuis le serveur entre les deux candidats, comme affiché dans l'application de la figure 8a. Notez que les URL et autres métadonnées fournies sont également potentiellement de longueur variable, et la longueur de l'URL est totalement imperceptible pour l'utilisateur.

Nous avons vérifié cette vulnérabilité en configurant un proxy entre notre application et notre serveur API et en enregistrant toutes les communications via tcpdump. Nous avons ensuite utilisé l'application pour participer deux fois à une élection , une fois pour le candidat « court » et une fois pour le candidat « long ». La figure 9 montre les tailles de texte chiffré résultantes en octets (en particulier, la longueur du champ TLS Application Data par paquet) dans les deux exécutions - dans les deux cas, le deuxième paquet (paquet n ° 1) correspond à la soumission réelle du vote, où le reste est un autre protocole divers requêtes impliquées dans le vote et la maintenance des utilisateurs. La longueur de ce paquet indique clairement quel candidat a été sélectionné, se distingue facilement des autres paquets dans le protocole et, surtout, sa taille n'est affectée par aucun paramètre qui varie selon l'utilisateur . crypter

13La taille du texte chiffré ne variera pas selon l'utilisateur, mais peut varient légèrement en fonction de l'implémentation TLS du téléphone.

La topographie exacerbe cette vulnérabilité. Dans l'implémentation de Voatz , les données sont compressées par gzip au niveau de la couche application avant d' être chiffrées via TLS, ce qui aurait pu offrir une certaine confidentialité , en supposant que la compression seule suffisait à masquer les différences de taille entre les textes en clair. Étant donné que Voatz crypte les données sortantes avant que le système n'applique gzip et que la compression d'une charge utile déjà cryptée ne réduira pas sa taille, cette étape est rendue immatérielle et la longueur du texte chiffré du paquet final est maintenue proportionnelle à la taille du texte en clair.
Le résultat est que (bien que les chiffres présentés ici ajoutent intentionnellement du texte pour exagérer l'effet à des fins pédagogiques), une différence modeste de quelques octets peut être suffisamment importante pour déterminer les préférences de l'électeur.

Pour que cette attaque fonctionne, nous faisons les deux hypothèses suivantes :

1. L'attaquant peut apprendre les options de vote présentées (peut-être par lui-même en votant et en accédant à la représentation JSON des options de vote).

2. Le serveur n'envoie pas d'une manière ou d'une autre les options de vote à l'appareil rembourré pour être de longueur égale.

La première hypothèse n'est probablement pas un problème compte tenu des attaques présentées au §5.1. Par exemple, un attaquant n'a qu'à être un électeur inscrit, avoir déjà exploité l'appareil d'un électeur inscrit et été témoin de ses options de vote, ou avoir surveillé d'une autre manière un électeur votant d'une manière particulière et enregistré le résultat.

La deuxième hypothèse est également susceptible de tenir, car nous ne trouvons aucune preuve que l'application se défend contre cette attaque - il n'y a pas de code pour supprimer les symboles ou les espaces superflus des questions de vote avant qu'elles ne soient présentées, et d'autres transactions qui impliquent des informations sensibles sur l'utilisateur. sont entièrement générés côté appareil et indépendants du serveur (comme le nom, l'âge et l'emplacement de l'utilisateur), et ne sont pas non plus rembourrés.
Enfin, si cette hypothèse ne tient pas, une version limitée de l'attaque est toujours viable : si l'utilisateur ne sélectionne aucun candidat et saute complètement la question, l'appareil envoie au serveur une liste vide.

Notez que ce canal auxiliaire permet à l'attaquant de détecter l' intention de l'électeur avant que le bulletin de vote n'arrive au serveur. Si l' attaquant est en mesure de bloquer les paquets en route vers le serveur (comme le ferait, par exemple, un FAI ou un propriétaire de réseau), l' adversaire pourrait intentionnellement supprimer ce paquet et empêcher de manière adaptative l'électeur de soumettre son bulletin de vote. Pour l'utilisateur, cela ressemblerait à une interruption de service du côté de Voatz et pourrait dégrader suffisamment l'expérience pour empêcher l'électeur de voter .

5.4 Autres observations et faiblesses

Confidentialité et préoccupations géostratégiques : l'application Voatz est intrusive pour la vie privée. Les informations envoyées à Voatz et/ou à des tiers associés à ce service incluent l'identité de l'utilisateur

e-mail, adresse physique, date de naissance exacte, adresse IP, une photo actuelle d'eux-mêmes, le modèle de leur appareil et la version du système d'exploitation, et la langue préférée. L'application demande également des autorisations pour lire le GPS de l'utilisateur lors de la première connexion, bien que nous n'ayons pas identifié exactement ce que l'application fait avec ces informations. Enfin, Voatz fait un usage intensif de code tiers (voir annexe B) ; Voatz comprend plus de 22 bibliothèques fournies par 20 fournisseurs différents.

L'une des utilisations signalées du logiciel de Voatz concerne les électeurs militaires à l'étranger, ce qui indique que les informations divulguées sur ses utilisateurs pourraient également fournir aux adversaires des informations sur les déploiements militaires américains. Notez que l'adresse IP de l'électeur peut à elle seule contenir des informations sur l'emplacement de l'utilisateur . Jumio, Crashlytics et Zimperium peuvent donc déduire les déploiements de troupes.

Sensibilité à la coercition : comme mentionné au point 4.2, l'application ne demande jamais à l'électeur de ressaisir son code PIN lors de la connexion après l'inscription, et ne semble pas indiquer à l'utilisateur si un bulletin de vote a été revoté ou annulé. Cela indique que l'application rend les utilisateurs vulnérables aux attaques par coercition. Considérez un électeur endormi ou autrement incapable. En supposant que l'attaquant ait un accès physique à l'appareil et à l'utilisateur, et que l'appareil soit déverrouillable via l'empreinte digitale de l'utilisateur, un attaquant aurait facilement la possibilité de voter au nom de l'utilisateur. Ce modèle de menace est très pertinent dans le cas de la violence conjugale [28, 54].

5.5 Reçu vérifié de l'électeur

D'après ce qui peut être discerné à partir de la documentation disponible et du code de l'application, il est très difficile de savoir ce que garantit le reçu de Voatz. En dehors de la fonction de demande de mot de passe mentionnée au §4.3, il n'y a aucune mention du reçu dans l' application ou son binaire, et il ne semble pas que l'application fournisse une méthode pour vérifier que le bulletin de vote a été compté dans la blockchain d'enregistrement - ou , au-delà de la documentation de Voatz, qu'une telle blockchain existe.

On ne sait pas non plus si le système de Voatz est E2E-V. À la connaissance des auteurs, les systèmes E2E-V dans la littérature de recherche exigent généralement qu'un électeur se rende dans un bureau de vote et utilise un bulletin de vote (par exemple Scantegrity [25] et StarVote [16]), une communication hors bande avant ou après l'élection (voir, par exemple, le vote par code [24] et Remotegrity [76]), ou un moyen d'effectuer des défis cryptographiques au moment de la soumission (voir Helios [10]).
En supposant que le PDF envoyé à l'utilisateur ne contienne pas de code en cours d'exécution, il serait difficile de déterminer comment le système pourrait éventuellement atteindre E2E-V, et, bien que la FAQ de Voatz semble vanter la vérifiabilité des électeurs, elle ne prétend pas explicitement être E2E-V.

Quoi qu'il en soit, la fourniture de tels reçus présente des difficultés pratiques importantes . Dans le cas où l'application présentait une sorte de vérification cryptographique concrète sans E2E V, cela pourrait permettre à l'utilisateur de prouver la façon dont il a voté - en violation des exigences de liberté de réception et de coercition résistance. Si le reçu arrive sous forme de PDF crypté, on ne sait pas comment Voatz peut prouver à l'utilisateur que le PDF crypté provient réellement de Voatz et, s'il est vérifié dans l'application, comment on protégerait le processus de vérification des attaques de modification de l'interface utilisateur . présenté au §5.

Enfin, il existe d'importants problèmes d'utilisation du récépissé qui nécessitent une analyse. De quelle solution un utilisateur dispose-t-il si le bulletin de vote soumis et le récépissé ne correspondent pas ? Comment un utilisateur sait-il quand attendre un accusé de réception ? Si le récépissé est envoyé ou retardé jusqu'à la post-certification de l'élection, n'y a-t- il pas de correction d'une erreur ? Comment inciter les électeurs à relever les défis nécessaires pour que le système de vérification soit efficace ? Nous notons en outre que bon nombre de ces questions sont enracinées dans des problèmes de recherche ouverts dans l' espace E2E-V [20].

La transparence dans la conception ici aiderait les responsables électoraux et les électeurs à comprendre ces compromis, et sans plus d'informations, une analyse complète de ces reçus n'est pas possible.

6 Discussion et conclusion

Divulgation responsable : Compte tenu de la sensibilité accrue entourant les questions de sécurité électorale et en raison de craintes de représailles potentielles, nous avons choisi d'alerter le Département américain de la sécurité intérieure (DHS) et de coordonner anonymement la divulgation par le biais de leur Cybersecurity and Infrastructure Security Agency (CISA). Avant d'annoncer publiquement nos conclusions, nous avons reçu la confirmation du fournisseur et, bien qu'il conteste la gravité des problèmes, il a semblé confirmer l'existence de la vulnérabilité du canal auxiliaire et des problèmes d'entropie du code PIN.14 Nous avons également parlé directement avec l'élection concernée. fonctionnaires dans le but de réduire le risque de nuire à tout processus électoral.

Bug Bounties en tant qu'outil de transparence et d'audit : comme mentionné précédemment, nous avons analysé la version la plus récente de l'application disponible dans le Google Play Store au 1er janvier 2020. Voatz propose également une version "bug bounty" de l'application via un troisième service de fête appelé HackerOne [5]. La société présente la prime aux bogues comme preuve de l'engagement de Voatz envers les audits indépendants, ainsi que la «vérification communautaire» du produit [6]. Nous avons choisi de ne pas examiner cette version de l'application pour plusieurs raisons.

Premièrement, l'évaluation de l'application Bounty seule introduirait des menaces supplémentaires pour la validité, et comme les différences entre cette version et celles qui ont été mises en service ne sont pas claires, nous avons choisi de pécher par excès de réalisme. Pire encore, toutes les applications sont indépendamment masquées de manière aléatoire, de sorte que l'analyse statique de chacune nécessite un long processus de désobscurcissement manuel.

14Le fournisseur a partagé des informations supplémentaires, mais, comme ces détails faisaient partie de communications confidentielles dans le cadre du processus de divulgation des vulnérabilités, ils ne sont pas inclus dans ce document. Rien de ce qui est fourni par le vendeur ne contredit les constatations factuelles de cet article.

répéter ce travail sur une deuxième application représente un effort supplémentaire important.

Deuxièmement, et surtout, la prime ne fournit aucun aperçu utile supplémentaire de l'infrastructure du serveur de Voatz, ni ne fournit aucune source ou binaire pour le serveur API à tester . En effet, lorsque la décision d'analyser l'application en direct a été prise, l'application Bug Bounty de Voatz et l'application Google Play n'ont pas réussi à se connecter.

Enfin, les termes de la prime de bogue contiennent des restrictions intenables qui entravent un dialogue ouvert sur le système. Par exemple, la prime de bogue exclut à la fois les attaques MITM et les attaques nécessitant un accès physique à l'appareil. Cette restriction d'accès physique pourrait être lue pour exclure toutes nos attaques sur l'appareil - Pour simuler un attaquant ayant accès à une vulnérabilité de niveau racine à distance, nous avons utilisé une technique manuelle de jailbreak qui nécessite un accès physique. La restriction MITM mettrait de la même manière l'attaque par canal latéral, ainsi que l'analyse d'un adversaire qui contrôle le serveur API de Voatz, explicitement hors de portée. Pire encore, la prime aux bogues, en coordination avec leur « politique de divulgation responsable », prive également les chercheurs d'un refuge sûr à moins qu'ils n'attendent pour divulguer leurs découvertes jusqu'à un moment arbitraire où Voatz décrète que le correctif du bogue doit être entièrement déployé [8].

En bref, la prime de bogue semble empêcher le chercheur de divulguer, ne fournit pas les ressources adéquates pour l'analyse et considère arbitrairement des classes entières de vulnérabilités réalistes en dehors de la portée de l'exercice. Nous concluons que la prime de bogue n'est pas particulièrement pertinente pour permettre aux chercheurs d'examiner, d'auditer ou d'améliorer la sécurité du système, et sert d'exemple de la façon dont de tels engagements peuvent ne pas être aussi efficaces qu'on pourrait l'espérer. Si l'objectif est de maximiser l'utilité des audits et d'augmenter la transparence grâce à une prime de bogue, les fournisseurs pourraient fournir le code source à la fois pour le serveur et le client, publier l'implémentation complète du système et les détails opérationnels, et libérer explicitement les chercheurs pour qu'ils divulguent leurs découvertes après l'industrie. - 90 jours standard ou, à tout le moins, selon un calendrier fixe et accessible au public.

Une note sur l'importance de la transparence : Le manque de sources publiques et la documentation incomplète exacerbent bon nombre des risques de sécurité et de confidentialité des informations documentés dans ce document, et servent d'exemple de l'importance de la transparence dans les logiciels électoraux. Bien que nous ayons dû consacrer beaucoup de temps et d'efforts pour désobscurcir l'application de Voatz et rendre les résultats accessibles pour l'analyse, les failles elles-mêmes ne sont pas nouvelles - les attaques par canal auxiliaire sont bien connues dans la littérature d'ingénierie et de recherche cryptographique, et de nombreux autres problèmes semblent être le résultat d'une mauvaise conception et d'une mise en œuvre non standard. Un accès ouvert à leur code, à la conception du système et à l'exécution des implémentations de test aurait probablement révélé ces failles rapidement et encouragé Voatz à les corriger, ou du moins à dissuader les responsables électoraux de mettre le public électoral en danger.

Il est également clair que le manque de transparence de Voatz n'a pas signifié entraver considérablement notre capacité à découvrir les failles présentées dans cet article, et échouera de la même manière à empêcher un adversaire disposant de ressources suffisantes de faire de même. Dans notre analyse, nous ne nous sommes jamais intentionnellement connectés aux serveurs de Voatz et avons recibler toutes les communications (y compris Crashlytics, Jumio et le serveur API de Voatz) vers notre propre infrastructure à la fois pour éviter de perturber leurs systèmes et pour nous conformer à la loi. Les criminels ou les agences de renseignement étrangères, en revanche, ne sont pas contraints de suivre la loi américaine et n'auraient probablement aucun scrupule à perturber les opérations normales, notamment en se connectant aux serveurs de Voatz ou en attaquant directement Voatz. Ces adversaires auront donc plus de facilité à découvrir les vulnérabilités exploitables et seront plus libres d'explorer les failles que nous n'avons pas pu investiguer ; il est possible que le backend de Voatz, l'infrastructure du serveur , la mise en œuvre de la blockchain et d'autres parties de leur service aient des problèmes impossibles à analyser sans autre accès.

Enfin, l'absence de divulgation explicite précisant exactement quelles informations sur les électeurs sont collectées, comment elles sont utilisées, combien de temps elles seront conservées et à quoi des tiers peuvent avoir accès constitue un écart important par rapport aux meilleures pratiques en matière de confidentialité, et est une omission particulièrement préoccupante étant donné la sensibilité des informations de vote. Comme mentionné au §4.2, la seule notification à l'utilisateur que Jumio existe est le logo pâle placé dans le coin inférieur droit de l'écran photo de l'application, et nous n'avons trouvé aucune indication accessible à l'utilisateur que Zimperium ou Crashlytics sont utilisés. Bien que la politique de confidentialité stipule que Voatz "peut transférer des informations personnelles à des tiers dans le but de fournir les services", elle ne divulgue jamais quelles informations ni à qui. Sans savoir où vont leurs informations personnelles, il ne peut y avoir de consentement éclairé - dans l'état actuel des choses, même la personne la plus diligente et la plus soucieuse de la vie privée est susceptible de mal comprendre et de supposer que ses données, en particulier ses informations d'identification, ne sont partagées avec Voatz.

Bien que Voatz ait une politique de confidentialité, son manque de transparence sur les pratiques de confidentialité importantes telles que le partage de données avec des tiers laisse les données des électeurs sans protection. En plus de servir d' avis aux consommateurs, les politiques de confidentialité sont un élément essentiel du cadre de protection de la vie privée, en particulier dans des juridictions telles que les États-Unis qui ne disposent pas de lois complètes sur la confidentialité ; la vie privée commerciale individuelle n'est généralement protégée aux États-Unis que si les entreprises prennent des engagements concrets dans leurs politiques de confidentialité déclarées [67]. Par exemple, étant donné que Voatz n'impose aucune limite explicite de durée de conservation des données sur Jumio dans une politique de confidentialité visible publiquement, les utilisateurs risquent de voir des informations sensibles liées aux élections conservées indéfiniment . Sauf restrictions statutaires locales et/ou obligations contractuelles inconnues des auteurs, l'absence d'une politique de confidentialité concrète rend Voatz et ses partenaires irresponsables de ces manquements à la vie privée, et ne permet pas de savoir si Voatz peut utiliser les informations en dehors du contexte de l'élection

Conclusion : En commençant par la Virginie-Occidentale, l'Utah et le Colorado , les États-Unis se sont aventurés sur la voie du vote par Internet.
Malgré les inquiétudes exprimées par les experts, une entreprise a vendu la promesse d'un vote mobile sécurisé, utilisant la biométrie, la blockchain et la cryptographie basée sur le matériel.

Pourtant, notre analyse a montré que cette application n'est pas sécurisée . Un adversaire passif du réseau peut découvrir le vote d'un utilisateur, et un adversaire actif peut perturber la transmission en réponse. Un attaquant qui contrôle l'appareil d'un utilisateur contrôle également son vote, écartant facilement les contre-mesures intégrées de l'application. Et notre analyse du protocole montre que celui qui contrôle le serveur a probablement tout le pouvoir d'observer, de modifier et d'ajouter des votes à sa guise.

Une question naturelle peut être pourquoi un tel service a été mis en service en premier lieu. S'adressant à la Harvard Business Review, le soutien de Voatz et philanthrope politique Bradley Tusk a déclaré :

> Ce n'est pas que les gens de la cybersécurité soient de mauvaises personnes en soi. Je pense que c'est qu'ils résolvent une situation, et je résous une autre. Ils veulent zéro risque technologique de quelque manière que ce soit. [...] Je résous le problème de la participation. [73]

Bien que nous apprécions et partageons le désir de Tusk d'augmenter la participation électorale, nous ne sommes pas d'accord pour dire que les risques de sécurité dans ce domaine sont négligeables ; nous pensons que les enjeux présentés dans ce travail l'emportent sur les gains potentiels de participation15. Comme nous l'avons montré dans cet article, les vulnérabilités de Voatz et les problèmes causés par un manque de transparence sont bien réels ; le choix ici n'est pas une question de participation, mais un adversaire contrôlant le résultat de l'élection et une perte de la vie privée des électeurs, mettant en cause l'intégrité de l'élection.

Compte tenu de la gravité des défaillances évoquées dans cet article, du manque de transparence, des risques pour la vie privée des électeurs et de la nature insignifiante des attaques, nous suggérons d'abandonner tout projet dans un avenir proche d'utiliser cette application pour des élections à enjeux élevés. Nous recommandons en outre que toutes les conceptions futures de systèmes de vote (et de systèmes connexes tels que les registres de vote électroniques) soient rendues publiques et que leurs détails, leur source, leur modèle de menace, ainsi que les processus sociaux et humains soient disponibles pour examen public.

Notez que toutes les attaques présentées dans ce document sont viables, quelle que soit la prétendue utilisation par l'application d'une blockchain, de biométries , d'enclaves matérielles et de réseaux mixtes. Nous nous joignons à d'autres chercheurs pour rester sceptiques quant à la sécurité fournie par les solutions de vote basées sur la blockchain [29, 41, 60], et du vote par Internet en général [58], et pensons que cela sert de leçon de choses en matière de sécurité - que le l'utilisation prétendue d'une série d'outils n'indique pas qu'une solution offre de réelles garanties de sécurité.

On ne sait toujours pas si un système de vote mobile ou Internet uniquement électronique peut pratiquement surmonter les exigences de sécurité strictes imposées aux systèmes électoraux. En effet, ce travail ajoute à la litanie de graves failles découvertes dans les approches uniquement électroniques et appuie la conclusion selon laquelle la norme actuelle - les systèmes indépendants du logiciel [61] utilisant des bulletins de vote papier vérifiés par les électeurs et des audits de limitation des risques [52] - reste l'option la plus sûre. Il incombe au développeur de prouver que son système est aussi sécurisé que ces méthodes bien approuvées , à la fois pour le public et la communauté de la sécurité, avant de pouvoir lui faire confiance en tant que composant crucial de la démocratie.
processus.

15En effet, il n'est pas clair si le vote mobile et par Internet augmente réellement la participation électorale. Une étude suisse [38] ne trouve, de manière quelque peu surprenante, aucune augmentation statistiquement significative de la participation électorale.

Post-scriptum

Une préimpression de cet article a été diffusée publiquement le 13 février 2020 et couverte par des articles de presse [64]. À la suite de nos découvertes, le comté de Mason, dans l'État de Washington, a annoncé qu'il cesserait d'utiliser Voatz, suivi rapidement par la Virginie- Occidentale [26].

Au lieu d'aborder les vulnérabilités signalées dans cet article, Voatz a répondu en attaquant la crédibilité de cette analyse. Dans un appel à la presse publique [9] et dans un article de blog intitulé "Voatz Response to Researchers' Flawed Report" [72], les responsables de l'entreprise ont minimisé la gravité des conclusions, contesté notre intention ainsi que la méthodologie globale de cet article, et a affirmé que nous avions examiné une version obsolète de l'application - mais curieusement, nous n'avons jamais nié les résultats eux-mêmes.

Le 13 mars 2020, Trail of Bits, une société de sécurité tierce, a publié un document détaillant une analyse de sécurité en boîte blanche de Voatz [3]. Leur analyse cite cet article, confirme la véracité et la gravité de toutes les conclusions rapportées ici et contredit explicitement la critique de Voatz - soutenant notre méthodologie en tant que processus standard de l'industrie et affirmant qu'il n'y avait pas de différences pertinentes en matière de sécurité entre l'application que nous avons examinée et l'application interne. maître. Trail of Bits a également confirmé que le code côté serveur contenait d'autres vulnérabilités opaques pour nous (trouver 48 problèmes au total) et que le protocole de Voatz n'est pas E2E-V, n'a trouvé aucune preuve du mixnet revendiqué par Voatz et a signalé que Zimperium était entièrement désactivé dans au moins un de leurs pilotes les plus récents. Enfin, HackerOne a depuis supprimé la prime de bogue de Voatz de sa plate-forme - une première pour l'entreprise - citant des inquiétudes concernant l'apparente incapacité de Voatz à interagir de bonne foi avec les chercheurs en sécurité

Malgré les conclusions de l'audit Trail of Bits (financé par Voatz), le PDG de Voatz continue de nier publiquement la véracité de nos conclusions, affirmant qu '«il y a tellement d'erreurs dans le rapport du MIT, qu'il est vraiment très difficile d'accepter ce rapport » [36].

Remerciements

Nous sommes éternellement reconnaissants envers l'équipe de la BU/MIT Technology Law Clinic dirigée par Andy Sellars, Tiffany C. Li et les étudiants John Dugger, Quinn Heath et Eric Pfauth. Sans les conseils, la patience et les efforts de cette équipe fantastique, ce document

n'aurait jamais été libéré. Nous tenons également à remercier Matt Blaze, Matt Green, Joseph Kiniry, Barbara Simons, David Jefferson, Neha Narula, Sunoo Park, Ron Rivest, Charles Stewart et Gerry Sussman pour leurs commentaires et leurs idées.

Michael Specter et Danny Weitzner sont soutenus, en partie, par l'Initiative de recherche sur les politiques Internet du MIT, et Spectre est en outre soutenu par la bourse Android Security and Privacy REsearch (ASPIRE) de Google. James Koppel a été soutenu par le Toyota Research Institute.

Les références

[1] Apktool. ibotpeaches.github.io/Apktool.

[2] Gestionnaire de Magisk. https://magiskmanager.com/.

[3] Notre rapport complet sur la plateforme de vote mobile Voatz | Blog Trail of Bits. https://blog.trailofbits.com/2020/03/13/our-full-report-on-the-voatz mobile-voting-platform/.

[4] Royaume. https://realm.io/.

[5] Voatz - Programme Bug Bounty. https://hackerone. com/voatz.

[6] FAQ sur Voatz. https://voatz.com/faq.html [https : //perma.cc/FBQ8-N875].

[7] Restez en sécurité avec la détection de vivacité 3D certifiée de Jumio. https://www.jumio.com/about/press releases/3d-liveness-detection/, 2018.

[8] Politique de divulgation des problèmes de sécurité de Voatz. https://blog. voatz.com/?p=1278, Août 2018. Catalogue de la bibliothèque : blog.voatz.com Section : Technologie.

[9] Voatz Open Press Call Transcrit du 13 février 2020, février 2020. Catalogue de la bibliothèque : blog.voatz.com Section : États-Unis.

[10] Ben Adida. Helios : vote d'audit ouvert basé sur le Web. Dans le symposium sur la sécurité USENIX, volume 17, pages 335 à 348, 2008.

[11] Andrew Yang. Moderniser le vote. https://www. yang2020.com/policies/modernize-voting/.

[12] Android. Système de magasin de clés Android. https: //developer.android.com/training/articles/keystore .

[13] Android. Paramètres.Secure | Développeurs Android . https://developer.android.com/reference/android/provider/Settings.Secure .

[14] Kévin Beaumont. Quelqu'un m'a envoyé un lien vers un autre compte Github, avec le nom de l'auteur répertorié sur Voatz. Il a un nom d'utilisateur codé en dur et mots de passe. https://twitter.com/GossiTheDog/statut/1026904510386585600 , Août 2018.

[15] Kévin Beaumont. Le site Web Voatz fonctionne sur une boîte avec SSH obsolète, Apache (plusieurs CVSS 9+), PHP etc. https://twitter.com/GossiTheDog/status/ 1026607447996354561, Août 2018.

[16] Susan Bell, Josh Benaloh, Michael D. Byrne, Dana De Beauvoir, Bryce Eakin, Philip Kortum, Neal McBurnett, Olivier Pereira, Philip B. Stark et Dan S. Wallach. STAR-Vote : un système de vote sécurisé, transparent, vérifiable et fiable. En 2013 Electronic Voting Technology Workshop/Workshop on Trustworthy Elections (EVT/WOTE 13), 2013.

[17] Susan Bell, Josh Benaloh, Michael D Byrne, Dana De Beauvoir, Bryce Eakin, Philip Kortum, Neal McBurnett, Olivier Pereira, Philip B Stark, Dan S Wallach, et al. Star-vote : un système de vote sécurisé, transparent, auditable et fiable . En 2013 Electronic Voting Technology Workshop/Workshop on Trustworthy Elections (EVT/-WOTE 13), 2013.

[18] Josh Benaloh. Élections simples vérifiables. TEV, 6 :5-5, 2006.

[19] Josh Benaloh. Assurance du scrutin via l'audit des bureaux de vote à l'initiative de l'électeur. EVT, 7:14–14, 2007.

[20] Matthew Bernhard, Josh Benaloh, J. Alex Halderman, Ronald L. Rivest, Peter YA Ryan, Philip B. Stark, Vanessa Teague, Poorvi L. Vora et Dan S. Wallach. Preuve publique des scrutins secrets. Dans Conférence conjointe internationale sur le vote électronique, pages 84–109. Springer, 2017.

[21] Matt Blaze, Jake Braun et Cambridge Global Advi sors. Village de piratage des machines à voter DEFCON 25. Actes du DEFCON, Washington DC, pages 1 à 18, 2017.

[22] Joseph A. Calandrino, Ariel J. Feldman, J. Alex Halderman , David Wagner, Harlan Yu et William P. Zeller. Examen du code source du système de vote Diebold. Université de Californie, Berkeley sous contrat avec le secrétaire d'État de Californie, 2007.

[23] D. Chaum. Reçus de vote secret : véritables élections vérifiables par les électeurs. IEEE Security Privacy, 2(1):38–47, janvier 2004.

[24] David Chaum. Surevot : aperçu technique. Dans Actes de l'atelier sur les élections dignes de confiance (WOTE'01), 2001.

[25] David Chaum, Richard Carback, Jeremy Clark, Alek sander Essex, Stefan Popoveniuc, Ronald L Rivest, Peter YA Ryan, Emily Shen et Alan T Sherman. Scant egrity II : vérifiabilité de bout en bout pour les systèmes d'élection par balayage optique utilisant des codes de confirmation à encre invisible. EVT, 8:1–13, 2008.

[26] Kévin Collier. La Virginie-Occidentale revient sur l'utilisation de l'application de vote par téléphone intelligent dans le primaire de l'État. Catalogue de la bibliothèque : www.nbcnews.com.

[27] Connie Loizos. Voatz a levé 7 millions de dollars en financement de série A pour sa technologie de vote mobile, juin 2019. http:// social.techcrunch.com/2019/06/ 06/voatz/.

[28] Ensoleillé Consolvo. Pratiques de confidentialité et de sécurité des personnes confrontées à la violence conjugale. 2017.

[29] David Jefferson, Duncan Buell, Kevin Skoglund, Joe Kiniry et Joshua Greenbaum. Ce que nous ne savons pas sur la « Blockchain » de Voatz Système de vote par Internet. https://cse.sc. edu/~buell/ blockchain-papers/documents/ WhatWeDontKnowAbouttheVoatz_Blockchain_.pdf , Mai 2019.

[30] S. Delaune, S. Kremer et M. Ryan. Résistance à la coercition et absence de récépissé dans le vote électronique. Dans 19th IEEE Computer Security Foundations Workshop (CSFW'06), pages 12 pp.–42, juillet 2006. ISSN : 2377-5459 .

[31] Commission d'assistance électorale. L'EAC publie le rapport annuel sur les dépenses en subventions, août 2017. https:// www.eac.gov/news/2017/08/16/eac releases-annual-grant-expenditure-report.

[32] Ittay Eyal et Emin Gün Sirer. La majorité ne suffit pas : le minage de Bitcoin est vulnérable. Dans Conférence internationale sur la cryptographie financière et la sécurité des données, pages 436–454. Springer, 2014.

[33] Ariel J. Feldman, J. Alex Halderman et Edward W. Felten. Analyse de sécurité de la machine à voter Diebold AccuVote-TS. 2006.

[34] David Fifield. Une meilleure bombe éclair. Au 13e atelier USENIX sur les technologies offensives (WOOT 19), 2019.

[35] Forrest Centi. Le projet pilote de vote mobile de Denver : un rapport. https://cyber-center.org/wp-content/uploads/2019/08/ Mobile-Voting-Audit-Report on-the-Denver-County-Pilots-FINAL.pdf, 2018.

[36] Lorenzo Emanuel Maiberg Franceschi-Bicchierai, Ja son Koebler. Une application de vote mobile déjà utilisée est remplie de défauts critiques, mars 2020. Catalogue de la bibliothèque : www.vice.com.

[37] Pierrick Gaudry. Briser le schéma de cryptage du système de vote par Internet de Moscou. arXiv preprint arXiv:1908.05127, 2019. https://arxiv.org/pdf/1908.05127.pdf .

[38] Micha Germann et Uwe Serdült. Vote et participation par Internet : témoignages suisses. Études électorales, 47:1–12, 2017.

[39] Dan Goodin. Le FBI dit aux utilisateurs du routeur de redémarrer maintenant pour tuer les logiciels malveillants infectant 500 000 appareils. https://arstechnica.com/information technology/2018/05/fbi-tells-router-users to-reboot-now-to-kill-malware-infecting 500k-devices/, Mai 2018.

[40] Dan Goodin. Le piratage de masse d'un routeur expose des millions d'appareils à un puissant exploit de la NSA, novembre 2018. https://arstechnica.com/information technology/2018/11/mass-router-hack exposes-millions-of-devices-to-potent nsa-exploit /.

[41] Rachel Goodman et J. Alex Halderman. In ternet Voting Is Happening Now and it Could Destroy Our Elections, janvier 2020. https: //slate.com/technology/2020/01/ internetvoting-could-destroy-our-elections.html .

[42] Thomas Haines, Sarah Jamie Lewis, Olivier Pereira et Vanessa Teague. Comment ne pas prouver que votre élection est sortie . Au 41e Symposium IEEE sur la sécurité et la confidentialité, 2019.

[43] Harvie Branscomb. Denver voatz sur les téléphones portables – examen initial. http://electionquality.com/2019/05/denver-voatz-1/ , Mai 2019.

[44] Jed Pressgrove. La tentative de piratage contre Voatz "n'est pas proche", disent les responsables. https://www.govtech.com/ security/The-Hack-Attempt-Against-Voatz Not-Close-Officials-Say.html, Octobre 2019.

[45] Jen Kirby. La Virginie-Occidentale teste une application de vote mobile pour les mi-mandats. Qu'est-ce qui pourrait mal se passer? https://www. vox.com/2018/8/17/17661876/west-virginia voatz-voting-app-election-security, Août 2018.

[46] Douglas Jones et Barbara Simons. Bulletins cassés : votre vote comptera-t-il ? Publications CSLI Stanford, 2012.

[47] Kévin Collier. Le FBI enquête sur une tentative de piratage présumée dans l'application de vote mobile. https:// www.cnn.com/2019/10/01/politics/fbi-hacking-attempt présumé-mobile-voting-app-voatz/index.html, Octobre 2019.

[48] Amy Klobuchar. S.1540 - 116th Congress (2019-2020): Election Security Act of 2019, mai 2019. https://www.congress.gov/bill/116th congress/senate-bill/1540/text.

[49] Tadayoshi Kohno, Adam Stubblefield, Aviel D. Rubin et Dan S. Wallach. Analyse d'un système de vote électronique. In IEEE Symposium on Security and Privacy, 2004. Actes. 2004, pages 27–40. IEEE, 2004.

[50] Larry Moore et Nimit Sawhney. SOUS LE CAPOT The West Virginia Mobile Voting Pilot, 2019. https: //www.nass.org/sites/default/files/2019- 02/white-paper-voatz-nass-winter19.pdf.

[51] Liat Weinstein. Des étudiants de l'Université du Michigan impliqués dans le piratage potentiel d'une application de vote. https://www.michigandaily.com/section/news briefs/university-michigan-students implicated-potential-voting-app-hack, 2019.

[52] Mark Lindeman et Philip B. Stark. Une initiation en douceur aux audits limitant les risques. Sécurité et confidentialité IEEE, 10(5):42–49, 2012.

[53] Sean Lyngaas. HackerOne coupe les liens avec la société de vote mobile Voatz après un affrontement avec des chercheurs, mars 2020.

[54] Tara Matthews, Kathleen O'Leary, Anna Turner, Manya Sleeper, Jill Palzkill Woelfer, Martin Shelton, Cori Man Thorne, Elizabeth F. Churchill et Sunny Consolvo. Histoires de survivants : Pratiques de confidentialité et de sécurité face à la violence conjugale. Dans Actes de la conférence CHI 2017 sur les facteurs humains dans les systèmes informatiques, pages 2189–2201. AMC, 2017.

[55] Maya Kosoff. "Une idée horriblement mauvaise": le vote par smartphone arrive, juste à temps pour les mi-parcours. https://www.vanityfair.com/news/2018/08/smartphone-voting-is-coming-just-in-time for-midterms-voatz, 2018.

[56] Lucas Méarian. Pourquoi le vote basé sur la blockchain pourrait menacer la démocratie. Computerworld, août 2019. https://www.computerworld.com/article/3430697/why-blockchain-could-be-a-threat to-democracy.html.

[57] Satoshi Nakamoto. Bitcoin : Un système de paiement électronique peer-to-peer. Rapport technique, Manubot, 2019.

[58] Ingénierie des Académies nationales des sciences et de médecine. Sécuriser le vote : protéger la démocratie américaine. The National Academies Press, Washington , DC, 2018.

[59] Robert W. Ney. HR3295 - 107e Congrès (2001- 2002) : Help America Vote Act of 2002, octobre 2002. https://www.congress.gov/bill/107th congress/house-bill/3295.

[60] Sunoo Park, Michael Spectre, Neha Narula et Ronald L. Rivest. Aller de mal en pis : du vote par internet au vote par blockchain. (BROUILLON).

[61] Ronald L. Rivest. Sur la notion d'« indépendance logicielle » dans les systèmes de vote. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 366(1881):3759–3767, 2008.

[62] David G. Robinson et J. Alex Halderman. Problèmes éthiques dans l'analyse de la sécurité du vote électronique. Dans Conférence internationale sur la cryptographie financière et la sécurité des données , pages 119–130. Springer, 2011.

[63] Ron Wyden. Lettre du sénateur Ron Wyden (D-Ore.) concernant Voatz. https://www.washingtonpost. com/context/sen-ron-wyden-d-ore-letter concernant-voatz/e9e6dd4f-1752-4c46-8e37-08a0f21dd042 /, novembre 2019.

[64] Matthieu Rosenberg. Voter sur votre téléphone : une nouvelle application électorale enflamme le débat sur la sécurité.

[65] Peter YA Ryan, David Bismark, James Heather, Steve Schneider et Zhe Xia. Prêt à voter : un système de vote vérifiable par l'électeur. IEEE transactions on information foren sics and security, 4(4):662–673, 2009.

[66] skylot. skylot/jadx, janvier 2020. https://github. com/skylot/jadx.

[67] Daniel J. Solove et Woodrow Hartzog. La ftc et la nouvelle common law de la vie privée. Colum. L. Rev., 114:583, 2014.

[68] Drew Springall, Travis Finkenauer, Zakir Durumeric, Jason Kitcat, Harri Hursti, Margaret MacAlpine et J. Alex Halderman. Analyse de la sécurité du système de vote par Internet estonien. Dans Actes de la conférence ACM SIGSAC 2014 sur la sécurité informatique et des communications, pages 703–715. AMC, 2014.

[69] Steven Rosenfeld. Compter Voatz : à l'intérieur de la technologie de vote la plus radicale d'Amérique. https://www.nationalmemo.com/counting voatz-inside-americas-most-radical-voting technology/, Mai 2019.

[70] Bennie G. Thompson. HR2660 - 116th Congress (2019-2020): Election Security Act of 2019, juin 2019. https://www.congress.gov/bill/116th congress/house-bill/2660/text.

[71] Voatz. Déclaration sur la lettre du sénateur Wyden, novembre 2019. https://blog.voatz.com/?p=1133.

[72] Voatz. Réponse de Voatz au rapport erroné des chercheurs. https://blog.voatz.com/?p=1209, Février 2020.

[73] Mitchell Weiss et Maddy Halyard. Voatz. Harvard Business Review, 2019. Étude de cas.

[74] Scott Wolchok, Eric Wustrow, Dawn Isabel et J. Alex Halderman. Attaquer le système de vote par Internet de Washington, DC. Dans Conférence internationale sur la cryptographie financière et la sécurité des données, pages 114–128. Springer, 2012.

[75] Yaël Grauer. Safe Harbor ou jeté aux requins par Voatz ? https://magazine.cointelegraph.com/2020/02/07/safe-harbor-or-thrown-to-the sharks-by-voatz/, Février 2020.

[76] Filip Zagórski, Richard T. Carback, David Chaum, Jeremy Clark, Aleksander Essex et Poorvi L. Vora. Remotegrity : Conception et utilisation d'un système de vote à distance vérifiable de bout en bout . Dans Conférence internationale sur la cryptographie appliquée et la sécurité des réseaux, pages 441–457. Springer, 2013.

[77] Kim Zetter. La Virginie abandonne enfin les "pires machines à voter" d'Amérique. Wired, août 2015. https: //www.wired.com/2015/08/virginia-finally drops-americas-worst-voting-machines/.

Un exemple JSON pour une soumission de vote

La figure 10 contient l'intégralité de la charge utile déchiffrée pour une soumission de vote et les paramètres renvoyés dans notre élection synthétique.

B Liste des tiers utilisés

Voatz utilise largement des bibliothèques tierces d'au moins 20 fournisseurs différents. Nous n'avons pas confirmé que toutes ces bibliothèques sont activement utilisées par l'application. De plus, une grande partie du code de Voatz est obscurcie, il peut donc y avoir d'autres bibliothèques utilisées dont nous n'avons pas connaissance.

• Jumio • Zimperium • Amazon AWS • Realm DB • Google Firebase / Crashlytics, gson, protobufs, zxking • Square OkHTTP & Retrofit • Datatheorem's TrustKit


```
{"voteDonnées" :
    [{
      "summary": "Meilleur       ,
      chat ?" " questionId
      " : "1", "isRCVFlag": faux
      "isRCV": faux     ,
      " description 1": "description fictive" [,
        déclarations   :
        {
          "summary": "Résumé de l'instruction",
          " statementId "  : "ID d'instruction",
          " description 3": " Description 3",
          "choix" : [ {

              "choixDétails " : {
                "imageUrl": " https : // bit . ly /36DJbC4",
                "webUrl": " https : // bit . ly /36DJbC4" },

              "choiceId" : "1", "
              description 1": "^" "     ,
              nonSelectable " : false "   ,
              description 2": "^" "     ,
              description "     : "Court"


          } ], " description 1": "Ceci est une sous−description" ,
          " description "  : "Ceci est une description de l'événement" ,
          "maxSelect": "1",
          "gender": "F", "
          description 2": " Description 2",
          "quartier "     : "Quartier de la déclaration"


      } ], " description 3": "desc bidon" ,
      " description 2": "desc bidon" "    ,
      description " : "desc bidon"


  } ], "auditToken" : "SomeAuditTokenValue",
  "controlNumber": "1",
  "customerId" : 267732387,
  "eventId " : 1 }
```


Figure 10 : Ce qui précède est l'intégralité de la charge utile déchiffrée pour une soumission de vote lors de notre élection synthétique.

• SoLoader et Fresco de Facebook • Relinker de Keepsafe • Bibliothèques knox de Samsung • Bibliothèques de capture de données de Microblink • Gestionnaire de préférences de Takisoft • MichaelRocks libphonenumber https://github.com/ MichaelRocks/libphonenumber-android • ReactiveX http://reactivex.io/ • Relex CircleIndicator https://github.com/ongakuer/CircleIndicator

• barre de progression du matériau de zhanghai https://github.com/zhanghai/MaterialProgressBar • JetBrains Anko https://github.com/Kotlin/anko • Joda.time https://www.joda.org/joda-time/ • Exploitation forestière de Jake Wharton https://github.com/ JakeWharton/timber • Les bibliothèques de polices de calligraphie de ChrisJenX https://github.com/chrisjenx/Calligraphie