Recommandations de l'ANSSI pour la sécurisation de StopCovid

Réf	Priorité (Majeure, Modérée)	Domaine (Architecture, Développement, Orga)	Recommandations
1	Majeure	Architecture	Cloisonner le backend de l'application, le service de génération/validation des QR codes et le service stopcovidpro ; assurer un filtrage réseau entre ces services.
2	Majeure	Architecture – backend	Segmenter le backend de l'application en plusieurs zones avec des rôles différents (front-office, middle-office, back-office, alerting/reporting/monitoring, development/deployment) ; assurer un filtrage réseau entre ces zones.
3	Majeure	Architecture – backend	Mettre en œuvre une protection anti-DdoS capable de contrer une attaque massive vis-à-vis du front-office exposé sur Internet ; le recours à un service spécialisé est recommandé.
4	Modérée	Architecture – backend	Nettoyer les méta-données non-utiles au protocole ROBERT (IP, ports, token d'authentification etc.) au niveau du frontal web
5	Majeure	Architecture – backend	Restreindre l'accès aux éléments cryptographiques au back-office
6	Majeure	Architecture – backend	Restreindre l'accès aux informations sensibles en clair au middle-office et au back-office (ex : clés de chiffrement et de signature, pseudonymes, informations de contact)
7	Modérée	Architecture – backend	Mélanger les entrées provenant des applications clientes au niveau du frontal web ; ne pas offrir au middle-office / back-office de vision agrégée d'un utilisateur
8	Majeure	Architecture – backend	Mettre en œuvre un coffre-fort électronique, matériel ou logiciel, pour protéger de manière robuste, sur le serveur central, les informations pseudonymisées envoyées par l'application stopcovid à partir du téléphone
9	Majeure	Architecture	Sécuriser l'accès des DevOps aux briques de déploiement : postes dédiés et durcis de préférence (pas de BYOD), accès aux API individuel, filtré, authentifié et journalisé
10	Majeure	Architecture	Mettre en œuvre des mécanismes d'audit de l'imputabilité et de la traçabilité des actions menées sur le système : - serveur de journalisation dédié, configuré en écriture seule - journaux signés et chiffrés (priorité modérée) - accessible à une autorité de contrôle

Réf	Priorité (Majeure, Modérée)	Domaine (Architecture, Développement, Orga)	Recommandations
11	Majeure	Architecture	Mettre en œuvre une collecte centralisée des journaux d'événements de sécurité sur l'infrastructure d'hébergement serveurs
12	Majeure	Architecture	Mettre en œuvre une stratégie de détection avec des alarmes reposant sur la collecte des journaux d'événements sur l'infrastructure d'hébergement.
13	Majeure	Architecture	Minimiser la surface d'attaque des serveurs centraux en procédant à la désactivation de tous les services inutiles.
14	Majeure	Architecture	Sécuriser l'accès au serveur de QR Code par les professionnels de santé protégeant la couche transport (IPsec de préférence) et en authentifiant les accès API entre le serveur de génération de QR codes et les postes clients des professionnels de santé (stopcovidpro)
15	Majeure	Architecture	Prévoir une procédure et des dispositifs de sauvegarde et de restauration
16	Majeure	Développement	Prévoir une procédure et des dispositifs pour le MCO/MCS des socles système et applicatif (l'hébergeur ne mettant à disposition que des images et pas de dépôts de mise à jour locaux)
17	Majeure	Orga	Mettre en œuvre une organisation de veille et de gestion des vulnérabilités pour maintenir un bon niveau de sécurité de l'application mobile et du serveur central durant toute la durée d'utilisation de l'application (veille des publications des vulnérabilités par les Internautes / qualification / correction / communication)
18	Modérée	Développement	Sécuriser le déploiement de code avec une phase préalable de revue de pairs
19	Modérée	Développement – Qrcode	Sécuriser la chaîne d'authentification et d'autorisation pour la génération d'un QR code par un professionnel de santé
20	Majeure	Développement – mobile	Sécuriser les communications avec le backend systématiquement en les encapsulant dans un canal TLS, dont les suites cryptographiques sont fixées explicitement, assurent la PFS et ce en respect du guide TLS de l'ANSSI
21	Majeure	Développement – backend	Sécuriser les communications des applications clientes et des postes des professionnels de santé systématiquement en les encapsulant dans un canal TLS, dont les suites cryptographiques sont fixées explicitement, assurent la PFS et ce en respect du guide TLS de l'ANSSI
22	Majeure	Développement – backend	Concernant le chiffrement des pseudonymes, mettre en œuvre de l'algorithme de chiffrement SKINNY-64/192.
23	Majeure	Développement – mobile	Mettre en œuvre du Certificate pinning (de préférence, offert par l'API Android) sur les applications clientes
24	Majeure	Développement – mobile	Protéger les données stockées localement (droits, chiffrement, effacement)
25	Modérée	Développement – mobile	Stocker ou wrapper le secret partagé au moyen des API qui permettent l'utilisation du composant crypto du téléphone (secure enclave pour iOS, hardware backed keystore pour android)
26	Modérée	Développement – mobile	Limiter dans le temps la rétention des informations éphémères
27	Majeure	Développement – backend	Limiter dans le temps la rétention des informations éphémères
28	Majeure	Développement – mobile	Ne pas utiliser de code natif (NDK) ou des bilbiothèques natives sur Android

Réf	Priorité (Majeure, Modérée)	Domaine (Architecture, Développement, Orga)	Recommandations
29	Majeure	Développement – mobile	Dépendances externes – S'assurer que qu'elles sont à jour des dernières versions disponibles
30	Majeure	Développement – mobile	Dépendances externes – choisir des bilbiothèques maintenues et respectueuses des présentes recommandations
31	Modérée	Développement – mobile	Dépendances externes – Réduire au strict nécessaire, pas de superflu
32	Majeure	Développement – mobile	Supprimer les dépendances au Play Services sur Android
33	Majeure	Développement – mobile	Supprimer l'usage des bilbiothèques Google (Cloud, Firebase, Crashlytics, …) sur Android
34	Majeure	Développement – backend	Dépendances externes – Mettre à jour des dernières versions disponibles
35	Modérée	Développement – backend	Dépendances externes – choisir des bilbiothèques maintenues et respectueuses des présentes recommandations
36	Modérée	Développement – backend	Dépendances externes – Réduire au strict nécessaire, pas de superflu
37	Majeure	Développement – mobile	Aucune information sensible dans les journaux en mode «release » (codes erreurs explicites, informations techniques relatives aux téléphones, etc.)
38	Majeure	Développement – backend	Aucune information sensible dans les journaux «release » (codes erreurs explicites, informations techniques relatives aux téléphones, etc.)
39	Majeure	Développement – mobile	Réduire au strict minimum nécessaire les permissions requises
40	Majeure	Développement – mobile	N'exporter aucun Service, aucune Activité, … sur Android à l'exclusion de l'Activité principale
41	Majeure	Développement – mobile	Désactiver les possibilités de debuggage en mode « release »
42	Faible	Développement – mobile	Supprimer tout code mort, obsolète, ou inaccessible
43	Majeure	Développement – backend	Supprimer tout code mort, obsolète, ou inaccessible
44	Modérée	Développement – mobile	Utiliser une source d'aléa dont la qualité permet l'usage pour des opérations cryptographiques.
45	Majeure	Développement – backend	Utiliser une source d'aléa dont la qualité permet l'usage pour des opérations cryptographiques.
46	Majeure	Développement – mobile	Recourir aux services offerts par le système d'exploitation plutôt que développer des mécanismes internes (e.g. prise de photo pour le QR Code) pour éviter l'usage de permissions inutiles
47	Majeure	Développement – backend	Contrôler et filtrer de manière stricte toutes les entrées externes (e.g. paramètres requêtes HTTP) avant tout usage
48	Modérée	Développement – mobile	Ne pas utiliser les mécanismes de sérialisation natifs du langage (e.g. java.io.Serializable en Java)
49	Modérée	Développement – backend	Ne pas utiliser les mécanismes de sérialisation natifs du langage (e.g. java.io.Serializable en Java)
50	Majeure	Développement – mobile	Ne pas utiliser de mécanisme d'obfuscation de code
51	Majeure	Développement – backend	Utiliser au maximum les mécanismes de sécurité offerts par le framework (e.g. Spring Security)
52	Modérée	Développement – backend	Vérifier les mécanismes de sécurité offerts par le framework (e.g. protection contre les CSRF), et le cas échéant, activer les mécanismes qui ne le sont pas par défaut (e.g. Content Security Policy)
53	Majeure	Orga	Tous les travaux menés dans le cadre du projet StopCovid seront publiés sous licence open source afin de garantir l'amélioration continue du dispositif et la correction d'éventuelles vulnérabilités.

Réf	Priorité (Majeure, Modérée)	Domaine (Architecture, Développement, Orga)	Recommandations
54	Majeure	Architecture	Mettre en place d'un dispositif de détection des cyberattaques pour réagir au plus tôt en cas de tentatives de compromission du système.
55	Majeure	Architecure	Utiliser des noms de domaines en .gouv.fr pour stopcovid et pour stopcovid-pro afin d'assurer la confiance des citoyens Cette recommandation est valable pour les éventuels sous-domaines relatifs : - au site Web d'information et de présentation du projet Stopcovid (ex : application.stopcovid.gouv.fr) - à l'api (ex : api.stopcovid.gouv.fr)
56	Majeure	Communication	Appliquer les bonnes pratiques pour la publication de l'application dans les Store - utiliser le nom "StopCovid France" - utiliser les "insignes" gouvernementaux pour inspirer la confiance (cf. Mariane, etc.)