LIVRE BLANC | FÉVRIER 2015

Conception d'une architecture CA Single Sign-On plus sécurisée

Utilisation des paramètres existants pour renforcer la sécurité de l'architecture

ca technologies

Table des matières

Résumé 3

Section 1 : 4
Importance de la sécurisation des sessions CA SSO

Section 2 : 5
Paramètres clés pour changer le comportement de CA SSO

Section 3 : 8
Conception d'une architecture offrant une sécurité maximale

Section 4 : 10
Conclusions

Section 5 : 11
Références

Résumé

Défi

Largement déployée à travers le monde, la solution CA Single Sign-On (CA SSO) assure la sécurisation et l'authentification unique pour une vaste gamme d'applications présentant des besoins de sécurité différents. Pour gérer les sessions utilisateur, CA SSO utilise plusieurs méthodes, notamment les cookies. Bien souvent, les administrateurs configurent la solution CA SSO de sorte qu'elle envoie ces cookies à une multitude de serveurs Web, y compris à ceux qui n'en ont pas l'utilité. La configuration des applications de telle sorte qu'elles envoient un cookie à une multitude de serveurs, introduit une faille dans l'architecture qui laisse le champ libre à un pirate informatique pour voler des cookies et relire les cookies de session dans le but de se faire passer pour un utilisateur authentifié.

Solution

CA SSO utilise une approche en attente d'être brevetée, à savoir le contrôle de session amélioré avec DeviceDNA™, qui permet de limiter le risque d'attaques par relecture de session. CA SSO inclut une série de paramètres qui permettent de renforcer la sécurité des sessions, notamment l'utilisation de cookies « hôtes uniquement », dont la seule fonction est d'être retransmis à l'hôte qui les a créés. Cette approche peut être mise en œuvre en complément d'une méthode d'authentification unique (SSO) interapplications au niveau des utilisateurs finaux, et permettre en parallèle aux agents de communiquer d'un domaine à l'autre au moyen d'un fournisseur de cookies central. Le fournisseur de cookies peut fournir une référence à usage unique pour une session stockée dans un magasin de sessions centralisé en vue de transmettre la session d'une application à l'autre.

Avantages

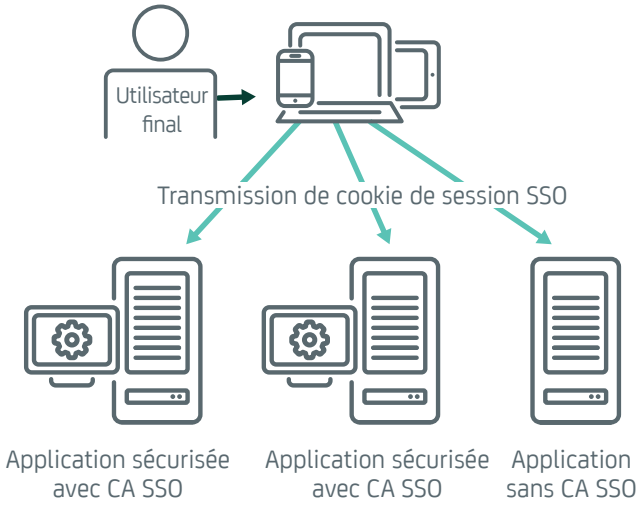
CA SSO permet aux administrateurs de configurer le comportement de l'architecture pour la totalité ou une partie de leurs applications. L'utilisation d'une architecture de type « hôte uniquement » permet de renforcer la sécurité en rendant plus difficile la capture de cookie de session, tout en limitant le risque de détournement de session auquel est exposé une application, jusqu'à ce que la session expire ou jusqu'à ce que la fonction de vérification de session détecte et arrête la session détournée.

Section 1 :

Importance de la sécurisation des sessions CA SSO

Le détournement de session n'est pas nouveau. Il a toutefois évolué et constitue désormais un risque de sécurité presque omniprésent, notamment depuis que le protocole HTTP 1.1 est devenu une norme, et ne se limite plus aux jetons de sessions CA SSO. La OWASP Foundation a relié le détournement de session aux attaques de type A2 « Gestion de sessions et authentification corrompues » de son classement des 10 risques de sécurité les plus critiques pour les applications Web. Si une session est détournée par un pirate informatique, ce dernier peut relire la session et afficher les informations des applications Web dans le contexte de l'identité volée. Par ailleurs, tous les fichiers journaux enregistreront les requêtes du pirate comme provenant d'un utilisateur dûment authentifié, ce qui rend l'attaque encore plus difficile à détecter.

Dans la plupart des déploiements, une session CA SSO est configurée en vue d'être partagée sur toutes les applications Web qui relèvent du même domaine de cookies (DNS) (par exemple un serveur Web sur le site ca.com). C'est le moyen le plus simple pour s'assurer que le cookie est envoyé à toutes les applications CA SSO concernées. C'est néanmoins aussi le moyen le plus vulnérable, étant donné que le navigateur Web fournit la session CA SSO aux applications qui en ont besoin, mais aussi à celles qui n'en ont pas besoin, toutes les applications partageant un même jeton.

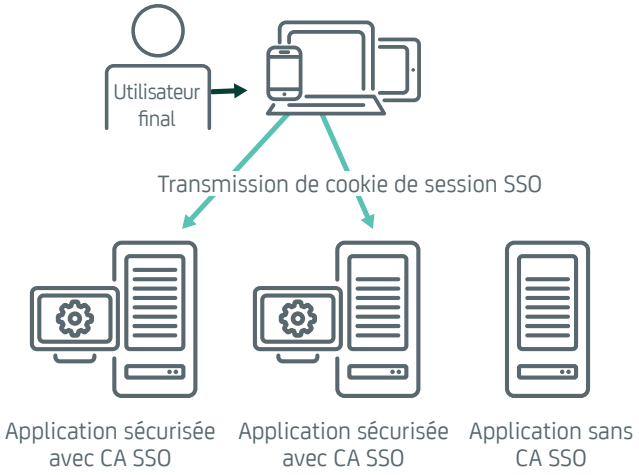


Section 2 :

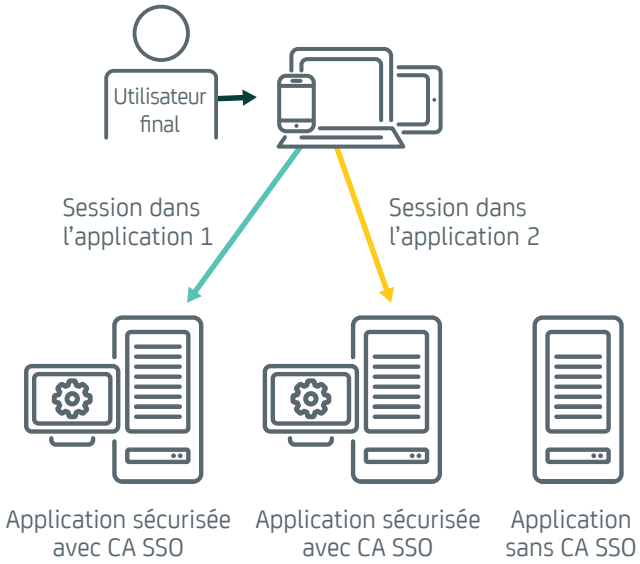
Paramètres clés pour changer le comportement de CA SSO

Pour élever le niveau de sécurité afin d'empêcher l'envoi de sessions aux applications qui n'en n'ont pas besoin, plusieurs paramètres peuvent être adaptés. Ceux-ci modifient le comportement général de CA SSO pour l'adapter à des scénarios plus sécurisés.

La première option possible consiste à envoyer le cookie CA SSO uniquement aux applications qui le nécessitent et non aux sites qui n'ont aucune raison d'afficher la session. Pour cela, il suffit de limiter la distribution du cookie à des hôtes spécifiques, au lieu de l'envoyer à tous les serveurs d'un même domaine.



CA SSO peut également être configuré de manière à utiliser des sessions spécifiques pour des applications particulières. Cela empêche d'autant plus l'utilisation d'une session détournée dans plusieurs applications.



Paramètres clés pour contrôler la sécurité des sessions CA SSO

Les paramètres suivants sont inclus dans la solution CA SSO depuis plusieurs années et permettent de modifier la configuration du comportement des agents et des passerelles. Ces paramètres sont tous disponibles dans l'objet de configuration de l'agent (ACO).

CookieDomain

Le paramètre CookieDomain permet de définir la valeur du domaine de cookies utilisée pour créer des cookies avec l'en-tête de réponse HTTP set-cookie. Il correspond par défaut à une chaîne vide, indiquant à l'agent que le domaine de cookie doit être dérivé de l'en-tête HTTP_HOST d'une requête basé sur le paramètre CookieDomainScope ci-dessous. La valeur « NONE » indique qu'aucune valeur ne doit être spécifiée pour le domaine de cookie. Cela définit un cookie de type « serveur uniquement ». Une valeur spécifique pour le domaine de cookie peut également être définie (par exemple, « .app.ca.com »). La méthode qui consiste à définir une valeur spécifique pour le domaine de cookie doit être utilisée avec précaution, car le domaine spécifié pour un cookie doit correspondre à une partie du domaine de la requête pour laquelle il est émis. Autrement dit, la valeur du domaine de cookie ne peut pas être arbitraire. Si un agent Web donné répond à des requêtes envoyées à plusieurs hôtes HTTP, AUCUNE valeur de domaine de cookie spécifique ne doit être utilisée ; dans les faits, ce type de valeur est rarement nécessaire.

CookieDomainScope

Le paramètre CookieDomainScope contrôle l'étendue d'une session en précisant la manière dont une valeur de domaine de cookie est dérivée de l'en-tête HTTP_HOST d'une requête. Par défaut, ce paramètre est défini sur 0. La valeur « 0 » indique la portée la plus large possible, autrement dit elle définit le domaine du cookie sur le domaine de premier niveau (par exemple, « ca.com »). La valeur « 1 » n'est pas autorisée, car « .com », « .net », etc., ne sont pas des domaines de cookie valides. La valeur « 2 » est identique à la valeur « 0 ». Les valeurs supérieures à 2 indiquent une portée plus précise lorsque le domaine de l'en-tête HTTP_HOST le permet. Par exemple, pour l'en-tête HTTP_HOST « monserver.sécurité.ca.com », les valeurs « 0 » ou « 2 » correspondent au domaine de cookie « .ca.com ». La valeur « 1 » n'est pas autorisée (le cas échéant, elle est ignorée et remplacée par la valeur par défaut « 0 »). Enfin, la valeur « 3 » correspond au domaine de cookie « .sécurité.ca.com ». La valeur « 4 » correspond à « monserveur.sécurité.ca.com ». Cependant, dans ce cas, il est préférable de définir le paramètre CookieDomain sur « NONE » comme indiqué précédemment. Lorsque le paramètre CookieDomain est défini sur NONE, le paramètre CookieDomainScope est ignoré, ce qui signifie que des cookies de type « serveur uniquement » doivent être utilisés. Dans le cas de cookies de serveur uniquement, l'étendue est TOUJOURS la valeur HTTP_HOST intégrale moins toute valeur de port indiquée.

CookieProvider

L'utilisation de cookies hôtes uniquement en parallèle d'une méthode d'authentification SSO entre les applications oblige à définir un site de fournisseur d'identités centralisé dont le rôle est de transmettre les informations de session aux autres applications. Cela correspond au paramètre CookieProvider de CA SSO. Ce paramètre désigne un serveur centralisé chargé de transmettre les informations sur les sessions aux autres applications Web distantes. Toute passerelle ou tout agent CA SSO peut endosser le rôle de fournisseur de cookies. Les agents qui utilisent ce CookieProvider utilisent l'URL du CookieProvider spécifiée dans le paramètre ACO.

EnableCookieProvider

Le paramètre EnableCookieProvider indique à une passerelle ou un agent SSO d'endosser le rôle de fournisseur de cookies. Il est recommandé de désactiver ce paramètre sur tous les ACO d'agents, à l'exception de celui désigné comme fournisseur de cookies. Cette manœuvre empêche un attaquant qui aurait détourné une session CA SSO sur une application d'élever ses privilèges afin de les utiliser pour accéder à d'autres applications.

StoreSessionInServer

Traditionnellement, les fournisseurs de cookies CA SSO inscrivent la session dans la chaîne de la requête HTTP dans le cadre d'une commande de redirection vers la cible finale. Cependant, le placement de ces données dans la chaîne de requête ouvre la porte de la session aux attaquants. Cette approche peut être remplacée au profit d'une autre dans laquelle le fournisseur de cookies CA SSO stocke la session dans un magasin de sessions centralisé, avant de transmettre une référence à usage unique à la session stockée dans la chaîne de requête. L'application demandant la session reçoit alors la session du serveur de stratégies avec lequel elle communique, et non directement de la chaîne de requête. Cette approche est très similaire au profil d'artefact SAML.

LimitCookieProvider

Lorsqu'un fournisseur de cookies centralisé est défini, ce dernier peut être utilisé pour créer de nouveaux cookies de session CA SSO pour les agents distants, ou pour autoriser un agent distant à créer une nouvelle session au niveau du fournisseur de cookies si l'utilisateur s'est authentifié directement sur le site distant. Ce paramètre peut forcer toutes les authentifications à se produire dans le domaine du fournisseur de cookies central et rejeter les sessions créées sur des applications distantes. L'utilisation de ce paramètre dépend de la stratégie de sécurité et d'entreprise mise en place. Si toutes les pages de connexion peuvent être centralisées à un même emplacement, il est recommandé d'utiliser ce paramètre.

TrackSessionDomain

Pour s'assurer qu'une session CA SSO n'est utilisée que pour le site auquel elle est destinée, utilisez le paramètre ACO TrackSessionDomain. Ce paramètre indique à l'agent Web de chiffrer et de stocker le domaine de la session dans le cookie de session. Lorsque d'autres requêtes lui parviennent, l'agent Web compare le domaine stocké dans le cookie au domaine de la ressource demandée. Si les domaines ne correspondent pas, l'agent Web rejette le cookie.

TrackCPSessionDomain

Un fournisseur de cookies CA SSO est chargé de transformer le domaine d'une session CA SSO en un autre domaine. Pour que cette transformation réussisse lorsque le paramètre TrackSessionDomain est activé, le fournisseur de cookies doit recevoir l'instruction de renommer le domaine au sein de la session afin qu'il puisse être utilisé à d'autres emplacements. Le paramètre TrackCPSessionDomain indique alors au fournisseur de cookies de valider le domaine du cookie déjà reçu avant de le transformer pour une autre application. Cette méthode empêche les pirates d'utiliser le fournisseur de cookies pour transformer les cookies arbitrairement d'un domaine en un autre (par exemple en envoyant au fournisseur de cookies un cookie « .app1.ca.com » intercepté qui deviendrait « .app2.ca.com »).

ValidTargetDomain

Le paramètre ValidTargetDomain identifie les domaines et hôtes valides pour les systèmes distants au cours du processus. Avant que l'utilisateur ne soit redirigé, l'agent compare les valeurs contenues dans l'URL de redirection aux domaines de ce paramètre. Sans ce paramètre, l'agent redirige l'utilisateur vers des cibles appartenant à un domaine quelconque. Ce paramètre permet d'éviter les attaques intersites au moyen d'un renvoi vers les pages d'authentification, les fournisseurs de cookies et les URL de vérification de session.

Section 3 :

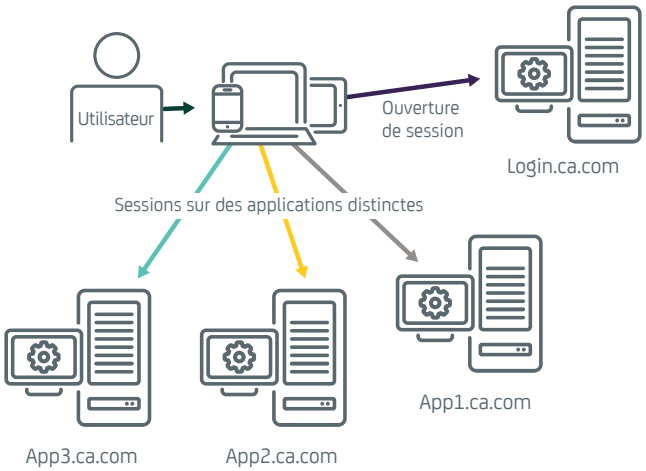
Conception d'une architecture offrant une sécurité maximale

L'utilisation de ces paramètres permet de mettre en place une architecture obligeant chaque application à ouvrir une session distincte qui lui est propre et exigeant une authentification des utilisateurs au niveau d'un point central, tout en maintenant la fonctionnalité SSO.

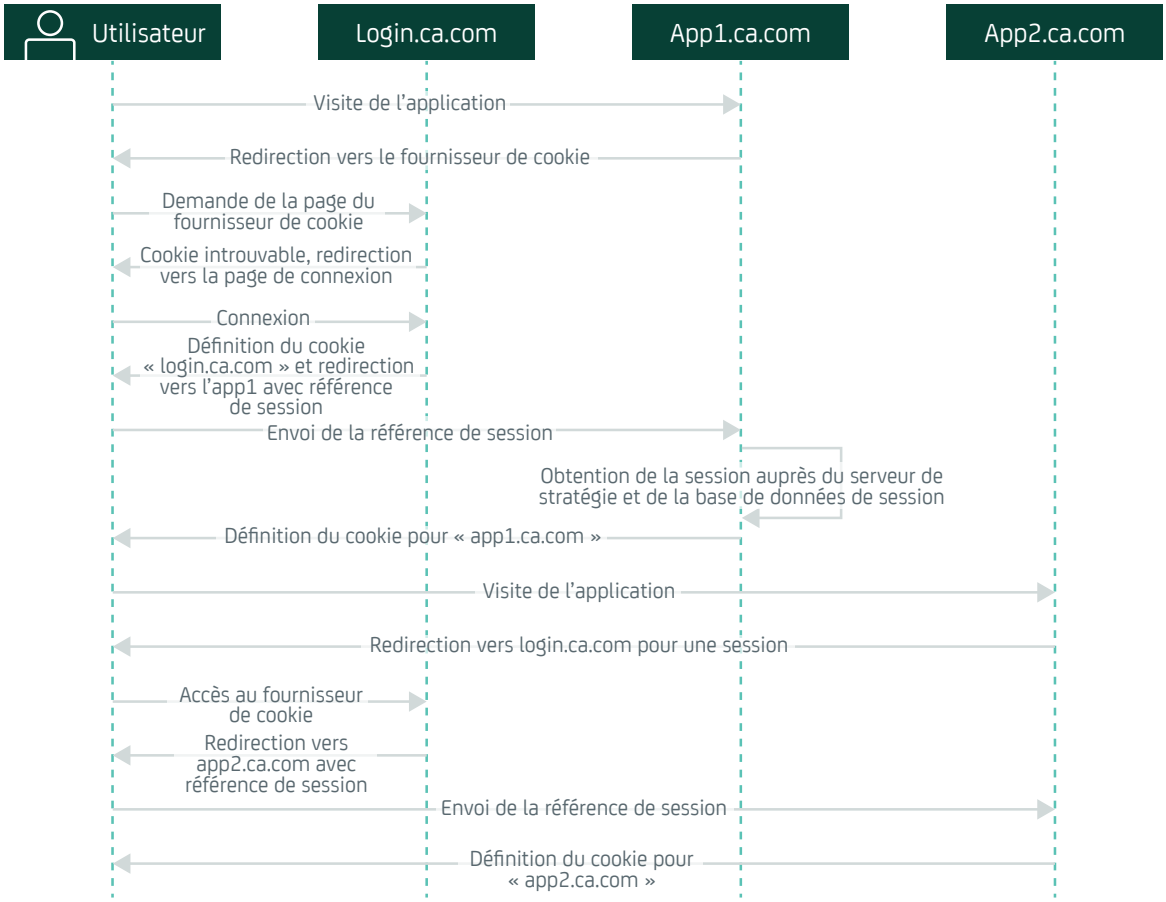
Dans l'exemple suivant, le site central login.ca.com fait office de fournisseur de cookies et héberge les pages d'authentification et plusieurs applications.

	Paramètre par défaut	Login.ca.com	App1.ca.com	App2.ca.com	App3.ca.com
CookieDomain	"" (chaîne vide)	NONE	NONE	NONE	NONE
CookieDomainScope	0 (utilisation du domaine de premier niveau)	Valeur par défaut	Valeur par défaut	Valeur par défaut	Valeur par défaut
CookieProvider		Valeur par défaut	https://login.ca.com/siteminderagent/SmMakeCookie.ccc		
EnableCookieProvider	Yes	Yes	No	No	No
StoreSessionInServer	No	Yes	Yes	Yes	Yes
LimitCookieProvider	No	Yes	No	No	No
TrackSessionDomain	No	Yes	Yes	Yes	Yes
TrackCPSessionDomain	No	Yes	Valeur par défaut	Valeur par défaut	Valeur par défaut
ValidTargetDomain	Tous les domaines ("")	App1.ca.com App2.ca.com App3.ca.com	Valeur par défaut	Valeur par défaut	Valeur par défaut

Les paramètres ci-dessus créent une architecture semblable à la suivante :



Une vue d'ensemble est représentée ci-dessous :

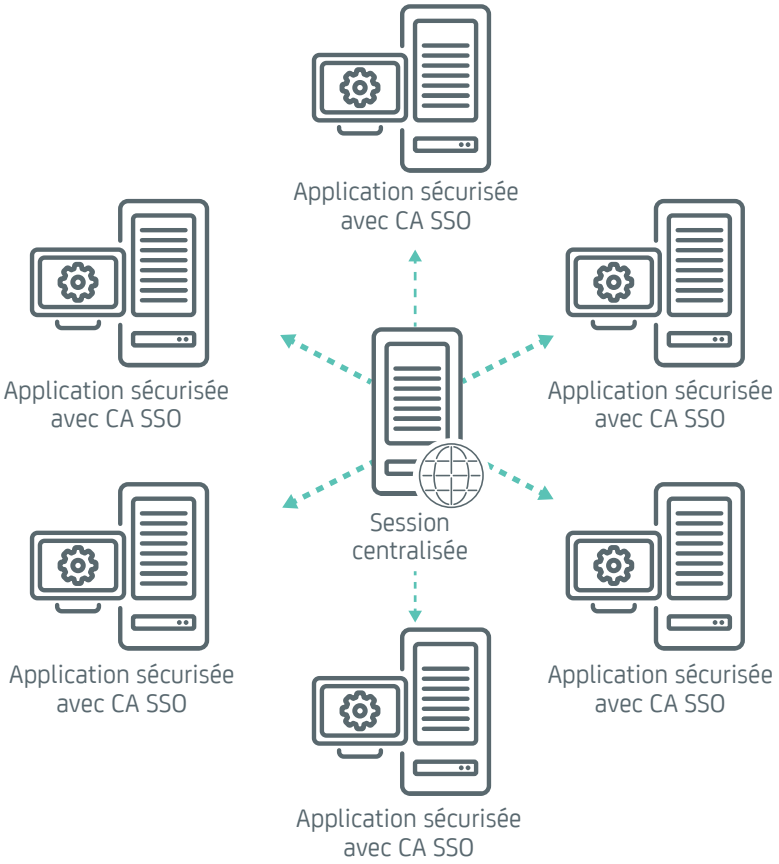


Cette architecture peut être combinée à d'autres mesures de contrôle des sessions CA SSO, comme l'utilisation de cookies « SSL uniquement » et « HTTP uniquement », le contrôle de session amélioré avec DeviceDNA pour obtenir les empreintes numériques des périphériques et l'application de stratégies d'expiration/d'inactivité de sessions adaptées. Une telle combinaison permet de verrouiller un environnement CA SSO afin de renforcer la sécurité des sessions, tout en maintenant la méthode SSO pour les utilisateurs finaux.

Section 4 :

Conclusions

Étant donné que les entreprises déploient des méthodes d'authentification multifacteurs avancées basées sur les risques, la sécurité des jetons de session fournis après l'authentification revêt une toute nouvelle importance, car ils constituent la faille logique suivante de l'infrastructure. La solution CA SSO permet de mettre en place une authentification unique et de gérer les sessions sur une multitude de sites à l'aide de cookies hôtes uniquement. Elle peut en outre utiliser les empreintes numériques des périphériques pour certifier que la session provient bien de l'hôte pour laquelle elle a été émise. La mise en place d'une architecture utilisant des cookies hôtes uniquement rend le détournement de session bien plus difficile en instaurant une topologie en étoile centralisée à la place d'une session unique pour l'ensemble du domaine.



Cette architecture limite également l'étendue des risques en cas de détournement de session. Étant donné que chaque application possède son propre cookie de session, en cas de détournement, le cookie de la session détournée ne peut être utilisé que pour l'application à laquelle il était destiné ; l'accès à toute autre application est rendu impossible jusqu'à ce que la vérification de session, le délai d'expiration ou d'autres contrôles invalident la session détournée.

Section 5 :

Références

Classement OSAWP : https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project

Section 6 :

À propos de l'auteur

Aaron Berman est Senior Advisor chez CA Technologies, notamment chargé de la stratégie commerciale et produit. Aaron possède plus de 15 ans d'expérience dans la résolution de problèmes, la conception, la mise en œuvre et la définition de stratégies pour les solutions de gestion des accès Web. Il a notamment contribué à la définition de tests de charge pour 100 millions d'utilisateurs sur CA Single Sign-On (anciennement CA SiteMinder) et CA Identity Manager (anciennement CA IdentityMinder), et géré plusieurs événements Federation Interop. Avant de rejoindre CA Technologies et de travailler pour Netegrity, Aaron dirigeait les services de support et d'avant-ventes pour les solutions de gestion des accès Web chez Raptor Systems/Axent Technology. Aaron a été vice-président et architecte principal de la branche Services de CA Technologies. Aaron est titulaire d'une licence en sciences informatiques de l'université de Syracuse, aux États-Unis.

Pour plus d'informations, rendez-vous sur **ca.com/fr/secure-sso**.

Restez connecté à CA Technologies sur ca.com/fr

CA Technologies fournit les logiciels qui aident les entreprises à opérer leur transformation numérique. Dans tous les secteurs, les modèles économiques des entreprises sont redéfinis par les applications. Partout, une application sert d'interface entre une entreprise et un utilisateur. CA Technologies aide ces entreprises à saisir les opportunités créées par cette révolution numérique et à naviguer dans l'Économie des Applications. Grâce à ses logiciels pour planifier, développer, gérer la performance et la sécurité des applications, CA Technologies aide ainsi ces entreprises à devenir plus productives, à offrir une meilleure qualité d'expérience à leurs utilisateurs et leur ouvre de nouveaux relais de croissance et de compétitivité sur tous les environnements : Mobile, Cloud, Distribué ou Mainframe. Pour en savoir plus, rendez-vous sur **ca.com/fr**.

1 Lhttps://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project

CS200-112938_0215