WHITE PAPER | 2015 年 2 月

セキュリティを強化する CA Single Sign-On Architecture の構築

既存の設定を使用したアーキテクチャのセキュリティ強化

ca technologies

目次

概要

課題

CA Single Sign-On（CA SSO）はセキュリティ・ニーズの異なる多様なアプリケーションを保護し、シングル・サインオンを提供するために世界中で広くデプロイされています。CA SSO ではさまざまな手法を使用してユーザのセッションを管理していますが、最も広く利用されているオプションはクッキーです。多くの管理者は CA SSO を設定して多様な Web サーバにクッキーを送信していますが、それには、セッション・クッキーにアクセスする必要のないサーバも含まれています。アプリケーションを設定してクッキーを多様なサーバに送信するとアーキテクチャが脆弱になり、攻撃者がセッション・クッキーを盗んで再生し、認証されたユーザになりすます可能性があります。

ビジネス・チャンス

CA SSO では、特許申請中の「Enhanced session assurance with Device DNA™」のアプローチを使用して、セッション・リプレイ攻撃を緩和できます。このソリューションには、セッションのセキュリティを強化するさまざまな設定がありますが、作成したホストにのみ返送される「host only」セッション・クッキーもその 1 つです。このアプローチでは、エンドユーザがシングル・サインオンで幅広いアプリケーションを利用できるだけでなく、中央のクッキー・プロバイダを使用してドメインを変更できますまた、このクッキー・プロバイダによって、セッションは一元化されたセッション・ストアに保存され、ワンタイム・リファレンスを使用して、他のアプリケーションに渡されます。

メリット

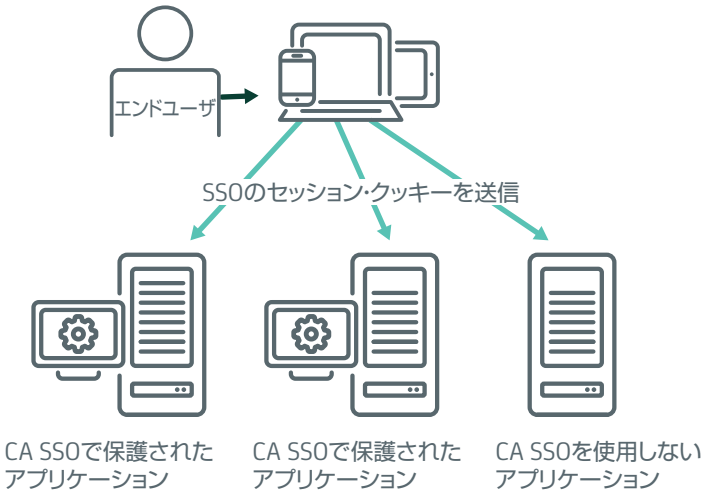
CA SSO では、すべてのアプリケーションまたはアプリケーションのサブセットに対するアーキテクチャの動作を設定できます。「host only」のアーキテクチャを使用すると、セッション・クッキーの盗聴が困難になり、盗聴された場合でも、悪用しようとしても特定のアプリケーションに制限されたままアイドル・タイムアウトやセッション保証機能によって停止されるため、セキュリティを強化することができます。

セクション 1:

CA SSO のセッションにおけるセキュリティの重要性

セッション・ハイジャックはクッキー・ハイジャックとも呼ばれますが、新しい脅威ではありません。HTTP 1.1 が標準になって以来、ほとんど永続的なセキュリティ・リスクになりましたが、これは CA SSO のセッション・トークンに限ったことではありません。OWASP Foundation では、セッション・ハイジャックは「トップ 10 最重要 Web アプリケーション・リスク」の中で A2 の「認証とセッション管理の不備」に該当するとしています。攻撃者が盗んだセッションを再生すると、Web アプリケーションからその盗んだアイデンティティに関する情報を入手できます。また、攻撃者のリクエストは有効な認証済みのユーザからのリクエストとしてすべてのログに記録されるため、攻撃の検出も極端にむずかしくなります。

多くの場合、CA SSO のセッションは、同じクッキー（DNS）のドメイン（ca.com の付く Web サーバなど）を共有するすべての Web アプリケーションで共有するよう設定されます。それによって、CA SSO を使用するすべてのアプリケーションへのクッキーの保存が簡略化されます。ただし、Web ブラウザによってすべてのアプリケーションに CA SSO のセッションが提供されると、すべてのアプリケーションが同じセッション・トークンを共有するため、リスクが最大化します。

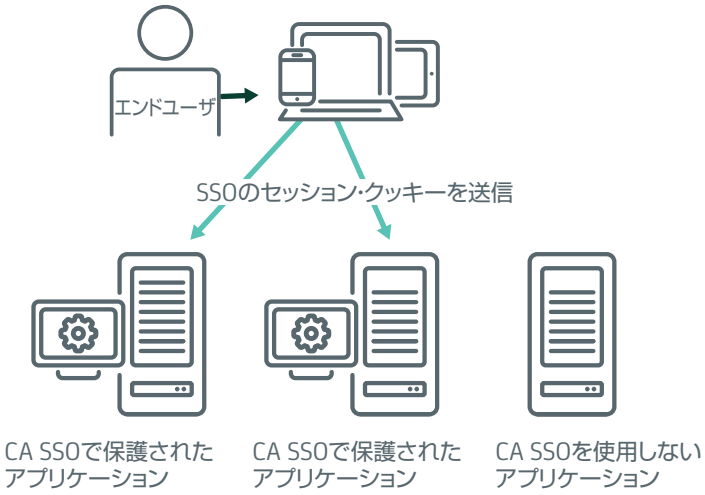


セクション 2:

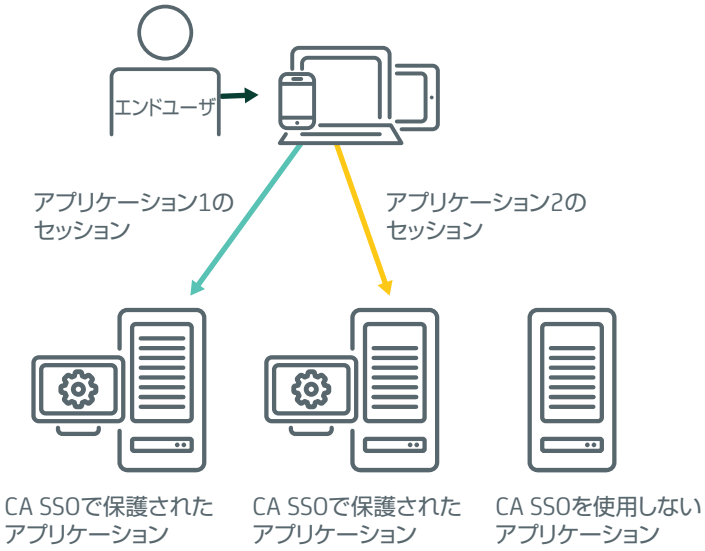
CA SSO の動作を変更するための主な設定

セキュリティ・レベルを強化して、必要のないアプリケーションにセッションが無作為に提供されるのを防ぐ設定はいくつかあります。これらの設定を使用すると、CA SSO の動作をより安全なシナリオに変更できます。

最初の変更では、必要なアプリケーションのみに CA SSO のクッキーが送信されます。それによって、セッションを必要としないサイトにクッキーが送信されるのを防ぐことができます。これは、クッキーを送信する範囲を同じドメインのすべてのサーバではなく、特定のホストに変更することで可能になります。



CA SSO では特定のアプリケーションに対して特定のセッションを使用するように設定することもできるため、セッションが盗まれても、複数のアプリケーションに使用されるリスクを回避できます。



CA SSO のセッションを保護する主な設定

ここでは、数年前から CA SSO で使用されているエージェントとゲートウェイの動作を変更する設定について説明します。これらの設定はすべて、Agent Configuration Object（ACO）にあります。

CookieDomain

CookieDomain の設定では、set-cookie を含む HTTP 応答ヘッダを使用してクッキーを作成するために、必要なクッキーのドメイン値を定義します。この設定をデフォルトの空の文字列のままにすると、以下で説明する CookieDomainScope の設定に基づいて、エージェントによってリクエストの HTTP_HOST ヘッダからクッキーのドメインが抽出されます。「NONE」の値は、設定するクッキーのドメイン値がないことを示します。この値によって、「server-only」クッキーが定義されます。また、クッキーのドメイン値を指定することもできます（「.app.ca.com」など）。クッキーのドメイン値を指定する場合、そのドメインはそのクッキーを発行するのに使用したリクエストのドメインの一部に一致する必要があります。任意の値はクッキーのドメイン値に使用できません。Web Agent で複数の HTTP ホストに対するリクエストを処理する場合、クッキーのドメイン値の指定が必要になることはほとんどないため指定する必要はありません。

CookieDomainScope

CookieDomainScope の設定では、リクエストの HTTP_HOST ヘッダから抽出するクッキーのドメイン値を定義して、セッション・スコープを制御します。デフォルト値は「0」です。この値はグローバルなスコープを意味し、クッキーの最上位のドメインが定義されます（「ca.com」など）。「.com」や「.net」などは正規のクッキーのドメインではないため、値「1」は使用できません。「2」は「0」と同じです。2 よりも大きい値では、HTTP_HOST のドメインで可能な範囲でさらにドメイン・スコープを定義します。たとえば、値「0」または「2」では、HTTP_HOST「myserver.security.ca.com」の「.ca.com」が定義されます。値「1」は使用できません（デフォルト値「0」が有効になり、無視されます）。値「3」では、「.security.ca.com」が定義されます。値「4」では、「myserver.security.ca.com」が定義されますが、その場合、上記で説明したように、CookieDomain を「NONE」に設定する方が適切です。CookieDomain を「NONE」に設定すると、CookieDomainScope は無視され、「server-only」クッキーが使用されます。server only クッキーの場合、スコープは常に HTTP_HOST の完全な値から任意に提供されたポートの値を取り除いた値になります。

CookieProvider

host only クッキーを使用し、複数のアプリケーション間でシングル・サインオンを可能にするには、一元化されたアイデンティティ・プロバイダのサイトで他のアプリケーションにセッションの情報を提供する必要があります。そこで、CA SSO の CookieProvider を使用します。これは、セッションの情報を他のリモートの Web アプリケーションに提供するために構築された一元化されたサーバです。CA SSO のゲートウェイまたはエージェントをクッキー・プロバイダとして使用できます。CookieProvider を使用するエージェントは、ACO の設定で指定された CookieProvider の URL を使用します。

EnableCookieProvider

EnableCookieProvider では、クッキー・プロバイダとして機能する SSO のゲートウェイまたはエージェントを指定します。指定したクッキー・プロバイダ以外のすべてのエージェントの ACO では、この設定をオフにすることをお勧めします。そうすることで、攻撃者が 1 つのアプリケーションの CA SSO セッションを盗んで他のアプリケーションへの侵入を試行した場合に、特権が昇格するのを防止できます。

StoreSessionInServer

従来の CA SSO のクッキー・プロバイダでは、最終的なリダイレクト先の一部として、セッションが HTTP クエリ文字列に挿入されます。このデータをクエリ文字列に挿入すると、攻撃者がセッションにアクセスするリスクが発生します。そのため、このアプローチの代わりに、CA SSO のクッキー・プロバイダを設定して、セッションを一元化されたセッション・ストアに保存してから、保存したクエリ文字列のセッションをワンタイム・リファレンスで参照できるようにします。そうすることで、セッションをリクエストするアプリケーションには、クエリ文字列から直接読み取る代わりに、接続しているポリシー・サーバからセッションが提供されます。このアプローチは、SAML 成果物のプロファイルによく似ています。

LimitCookieProvider

一元化されたクッキー・プロバイダを使用すると、そのクッキー・プロバイダを使用してリモート・エージェントのために新しい CA SSO のセッション・クッキーを作成したり、ユーザが直接リモート・サイトにログインしている場合は、リモート・エージェントによってそのクッキー・プロバイダで新しいセッションを作成できます。この設定によって、すべての認証が中央のクッキー・プロバイダのドメイン内で行われるようになり、リモート・アプリケーションで作成されたセッションは拒否されます。この設定の使用は、セキュリティ・ポリシーとビジネス・ポリシーによって決まります。すべてのログイン・ページに一元化された場所を使用できるなら、この設定を使用することをお勧めします。

TrackSessionDomain

CA SSO のセッションを目的のサイトでのみ使用するためには、TrackSessionDomain ACO の設定を使用します。この設定では、Web Agent によってセッションの目的のドメインが暗号化され、セッションのクッキー内に保存されます。その後のリクエストでは、リクエストされたリソースのドメインがセッションのクッキーに保存された目的のドメインと比較されます。ドメインが一致しない場合、そのセッションのクッキーは拒否されます。

TrackCPSessionDomain

CA SSO のクッキー・プロバイダの重要な機能は、CA SSO のセッションのドメインを変更することです。その変更を正しく機能させるために、TrackSessionDomain を使用して、クッキー・プロバイダにセッション内のドメインの名前を他の場所で使用できる名前に変更させます。また、他のアプリケーションのために変更する前に、そのクッキーのドメインをクッキー・プロバイダに検証させます。そうすることで、攻撃者がクッキー・プロバイダを使用してクッキーのドメインを変更するのを防止します（盗んだ「.app1.ca.com」のクッキーをクッキー・プロバイダに送信して、「.app2.ca.com」に変更するなど）。

ValidTargetDomain

ValidTargetDomain パラメータでは、処理中のリモート・システムの有効なドメインとホストを指定します。ユーザがリダイレクトされる前に、リダイレクトされる URL の値とこのパラメータのドメインがエージェントによって比較されます。このパラメータがないと、そのドメインがどのようなドメインであっても、ユーザはエージェントによってリダイレクトされます。この設定を使用すると、ログイン・ページ、クッキー・プロバイダおよびセッション保証の URL へリダイレクトされるため、クロスサイト攻撃を防止できます。

セクション 3:

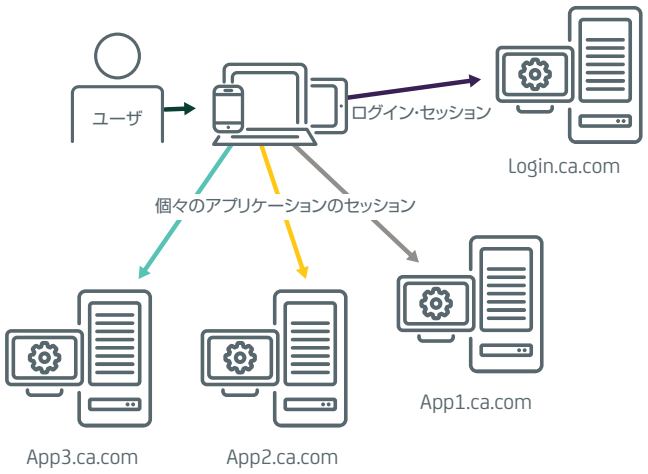
セキュリティを最大化するアーキテクチャの構築

これらの設定を使用すると、各アプリケーションに対してそのアプリケーションでのみ使用できるセッションが個別に割り当てられ、すべてのユーザに中央での認証が求められ、SSO の機能も維持されます。

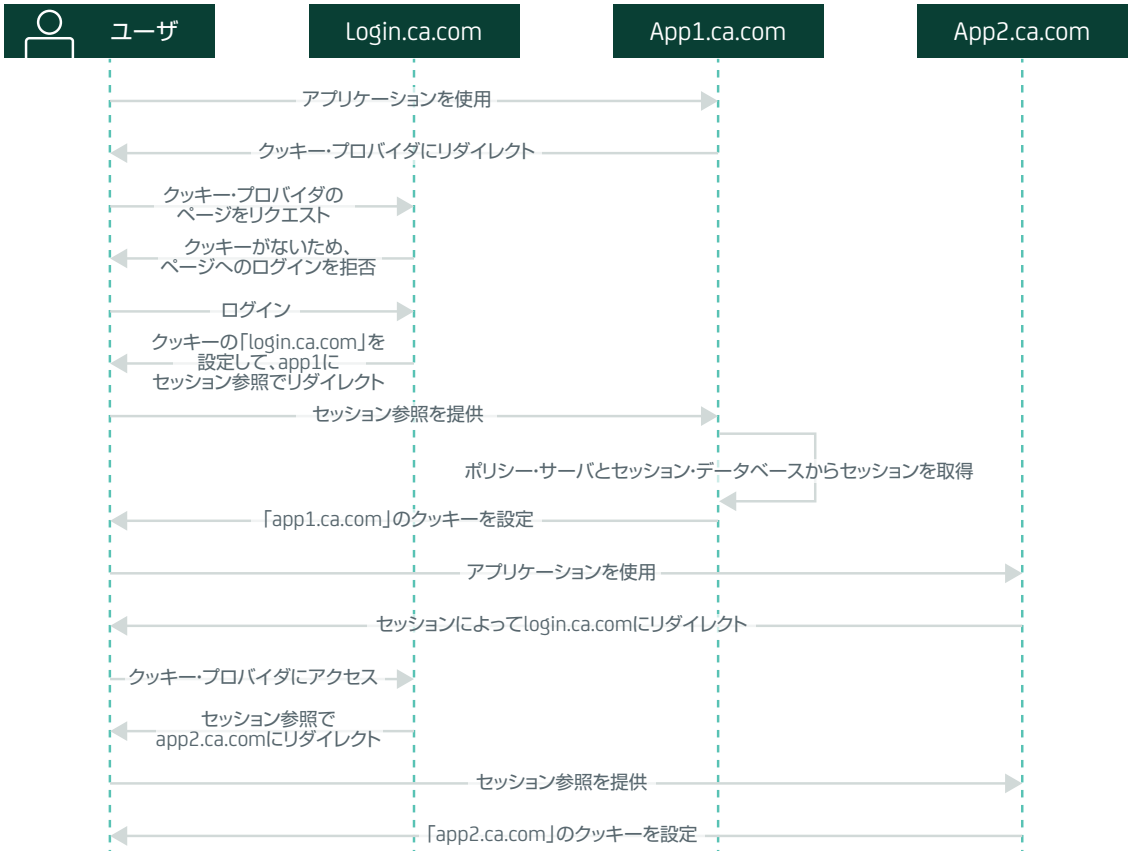
この例では、クッキー・プロバイダとして機能する中央のサイト「login.ca.com」によって、複数のログイン・ページとアプリケーションをホストしています。

	デフォルト設定	login.ca.com	app1.ca.com	app2.ca.com	app3.ca.com
CookieDomain	""（空の文字列）	NONE	NONE	NONE	NONE
CookieDomainScope	0（上位のドメイン・スコープ使用）	デフォルト	デフォルト	デフォルト	デフォルト
CookieProvider		デフォルト	https://login.ca.com/siteminderagent/SmMakeCookie.ccc		
EnableCookieProvider	はい	はい	いいえ	いいえ	いいえ
StoreSessionInServer	いいえ	はい	はい	はい	はい
LimitCookieProvider	いいえ	はい	いいえ	いいえ	いいえ
TrackSessionDomain	いいえ	はい	はい	はい	はい
TrackCPSessionDomain	いいえ	はい	デフォルト	デフォルト	デフォルト
ValidTargetDomain	すべてのドメイン（""）	app1.ca.com app2.ca.com app3.ca.com	デフォルト	デフォルト	デフォルト

上記の設定では、以下のようなアーキテクチャになります。



全体の概要は、以下のとおりです。

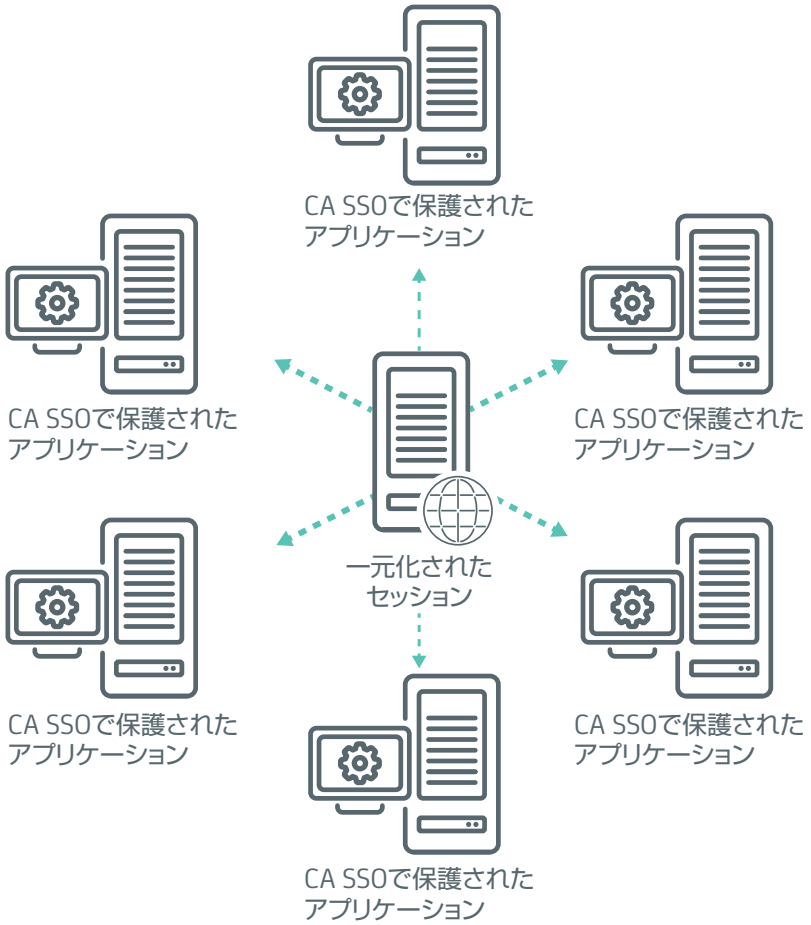


このアーキテクチャは、SCA SSO のセッションの他の制御機能（SL Only や HTTP Only のクッキーなど）と Enhanced Session Assurance with DeviceDNA（デバイスのフィンガープリンティングと適切なアイドル / セッション・タイムアウトのポリシー）を組み合わせることによって、CA SSO の環境をロックダウンしてセッションのセキュリティを強化しながら、エンドユーザに SSO を提供することができます。

セクション 4:

まとめ

多くの企業がリスクベースの高度な多要素認証システムをデプロイしていますが、認証後のセッション・トークンはインフラストラクチャにおける論理の脆弱性となるため、そのセキュリティを確保することは今後ますます重要になります。CA SSO では、host-only クッキーによって多様なサイトのシングル・サインオンとセッションを管理できるだけでなく、デバイスのフィンガープリンティングによって適切なホストからのセッションであるかを検証することもできます。host-only クッキーを使用するアーキテクチャでは、セッション・クッキーの盗聴を防止するため、ドメイン全体で 1 つのセッションを共有する代わりに、一元化されたスター型トポロジを使用します。



また、このアーキテクチャでは、セッションの攻撃を防止するために、リスクの原因も制限されます。各アプリケーションには個別のセッション・クッキーが割り当てられるため、セッション・クッキーが盗聴されても指定された 1 つのアプリケーションにしか使用できません。そのため、盗聴されたセッションは、他のアプリケーションのアクセスに使用されないままセッション保証、タイムアウトまたはその他の制御機能によって無効になります。

セクション 5:

参考資料

OSAWP のトップ 10:https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project

セクション 6:

著者について

Aaron Berman は現在、CA Technologies の現場組織のシニア・アドバイザとして製品および販売戦略のほか、製品管理の延長として、メッセージングなどの業務にも携わっています。Web アクセス管理ソフトウェアのトラブルシューティング、設計、導入、戦略で 15 年以上の経験がありますが、その豊富な経験には、CA Single Sign-On (旧 CA SiteMinder) と CA Identity Manager (旧 CA IdentityMinder) の 1 億回に及ぶユーザの負荷テスト、連携による相互運用性のさまざまなイベントの管理も含まれます。CA Technologies に入社する前は、Netegrity で Raptor Systems/Axent Technology の Web アクセス管理ソリューションのサポートとプレセールス・サービスを統括していました。また、CA Technologies のサービス組織で副社長兼設計責任者を務めた経験もあります。シラキュース大学でコンピュータ・サイエンスを専攻し、理学士号を取得しています。

詳細については、**ca.com/jp/secure-sso** をご覧ください。

ca.com/jp/でCA Technologiesにアクセスしてください

CA Technologies（NASDAQ:CA）は、企業の変革を推進するソフトウェアを作成し、アプリケーション・エコノミーにおいて企業がビジネス・チャンスを獲得できるよう支援します。ソフトウェアはあらゆる業界であらゆるビジネスの中核を担っています。プランニングから開発、管理、セキュリティまで、CA は世界中の企業と協力し、モバイル、プライベート・クラウドやパブリック・クラウド、分散環境、メインフレーム環境にわたって、人々の生活やビジネス、コミュニケーションの方法に変化をもたらしています。詳細については **ca.com/jp** をご覧ください。

CS200-112938_0215