5ZC-02

Webサービスのセッション窃取攻撃耐性の評価

川合 健太 †
名古屋大学大学院 情報学研究科 †

嶋田 創 ‡
名古屋大学 情報基盤センター ‡

1 はじめに

多要素認証の普及により，認証情報窃取よりもセッション窃取の方が攻撃者にとってコスト対性能比が良くなる傾向にある．Web において認証通過後のセッションは，Cookie としてブラウザに持たせる方式が一般的であるが，単純な実装ではセッション ID を窃取して送出すれば認証通過後のセッションが利用可能となる．セッション窃取耐性を高めるために，Fingerprint の利用やセッション ID の寿命を短くする手法が知られている．これらの手法が Web サービスに取り入れられているか，定期的に調査がされるのが望ましい．そこで，多要素認証に対応したいくつかの代表的な Web サービスのセッション窃取耐性の評価を行い，サービスの頑健性を検証した．具体的には，複数の Web サービスを対象にブラウザの拡張機能を用いて Cookie や User-Agent を操作し，セッション窃取に必要な Cookie の分析やウェブブラウザの情報をセッション情報として利用しているか検証した．

2 利用するブラウザの拡張機能

Cookie とはサーバがブラウザに送信するデータであり，その後のブラウザから同一ドメインへのリクエストの際に自動的に送信される．この情報を用いて，セッション管理やパーソナライズ，トラッキングが行われている．Cookie はクライアントが編集可能であり，本実験では Cookie-Editor [1] と呼ばれる拡張機能を用いた．Cookie-Editor を用いることで Cookie の編集，インポート，エクスポートが行える．

現代のブラウザにはクライアント側のストレージとして，Cookie の他に Web Storage が提供されている．具体的には Session Storage と Local Storage の２つが存在しており，Web StorageAPI からアクセスできる．本実験では Web Storage を操作する良いブラウザ拡張機能が見つからなかったため，確認・削除のみを各ブラウザが提供する開発者ツールによって行った．

User-Agent とはクライアントが Web サイトを訪問した際に使用中のブラウザや OS などの情報であり，ブラウザによって HTTP ヘッダの中に自動的に含まれるようになっている．Web サービスによってはこれを認証情報として利用している場合があり，異なるブラウザや OS からのアクセスを検知してユーザに通知したり，アクセスを保護する場合がある．User-Agent を改ざんするために Firefox では *MyBrowserAddon* が，Edge では *add0n.com* が提供する拡張機能 [2] を利用した．

3 実験

認証サービスを提供している A 社，B 社のアカウントサービスと大手 EC サイトの C 社，D 社の 4 つの Web サービスを対象にセッション窃取耐性の評価を行った．

はじめに同一の Google Chrome 上で Cookie の削除・復元を行った．各種 Web サービスへのログイン時に発行された Cookie を退避させた後にそれらをすべて削除し，ログイン状態を確認した．その後退避した Cookie を復元し，ログイン状態を確認した．さらに Cookie の復元によってログイン状態を復元できた Web サービスに対しては，

Assessment of Resistance Against Session Hijacking Attacks on Web Services
† Kenta Kawai, Graduate School of Informatics, Nagoya University
‡ Hajime Shimada, Information Technology Center, Nagoya University

表1　各種 Web サービスの認証に関連すると思われる情報の操作結果

サービスプロバイダ	同一ブラウザ上の Cookie 復元	認証に関する Cookie の洗い出し	異種ブラウザ間の Cookie 移動	異種ブラウザ間の Cookie 移動＋User-Agent 偽造
A 社	×	×	×	×
B 社	○	×	×	×
C 社	○	○ (2 個が該当)	○	○
D 社	○	○ (1 個が該当)	○	○

Cookie を変えながら削除・復元を繰り返し，発行された Cookie のうちセッションに関連すると思われるものを洗い出した．

続いて，異種ブラウザ間の Cookie 移動を行った．Google Chrome 上で各種 Web サービスにログイン時に発行された Cookie を Firefox，Edge にインポートし，ログイン状態を確認した．その後，上述の実験において洗い出したセッションに関連すると思われる Cookie に限定して同様の操作を行い，ログイン状態を確認した．

最後に User-Agent 偽造を行った状態で異種ブラウザ間の Cookie 移動を検証した．Firefox, Edge の双方で User-Agent を偽造し，Google Chrome 上での表示と一致させた．その後，先ほどと同様に異種ブラウザ間の Cookie 移動を行った．

4　実験結果

表 1 に実験結果をまとめたものを示す．認証サービスを展開する A 社と B 社は高い頑健性を持つ．とくに A 社においては同一ブラウザ上の復元でも再認証が要求され，強力な耐性を持つ．対照的に大手 EC サイトの C 社と D 社ではセッション窃取が有効であり，具体的にどの Cookie がセッション情報として利用されているかも明らかとなった．大手 EC サイトでは再認証によるユーザの購買意欲低下を避けるために，再認証が起きにくい仕様とした可能性も考えられる．

B 社は Cookie をすべて削除した状態でもいくつかのサイトをリダイレクトした後に，認証後ページに遷移することができた．また Local Storage の情報を削除することでログアウトされたことから，認証やセッションに関する情報を Local Storage に保持させている可能性がある．D 社では Session Storeage に情報を保持させており，再認証の際にはログインボタンをクリックするだけよい．

User-Agent 偽造の有無によって結果が変わったサービスは存在せず，セッション維持への影響は本実験では確認できなかった．

5　まとめと今後の課題

本論文では大手 Web サービスの Cookie の復元・移動や User-Agent を操作することによってセッション窃取耐性の検証を行った．Web サービスによっては Web Storage を用いた認証やセッション保持がおこなわれている可能性がみられた．これらの活用によってユーザの負担を軽減していると考えられるが，その頑健性については検証する余地がある．また本実験は同一 PC 上で行った．異なる端末やネットワークを介して実際にセッション窃取を行い，セッション窃取が成功するかを将来の検証課題とする．

本論文は脆弱性に関する研究では無いが，研究倫理の点より論文公開によるネガティブな影響を考慮し，サービス名が推測できない形で執筆した．

参考文献

[1] Cookie-Editor https://cookie-editor.com/ (参照 2023/11/07)
[2] User-Agent Switcher::MyBrowserAddon https://mybrowseraddon.com/useragent-switcher.html (参照 2023/11/27)
[3] User Agent Switcher and Manager https://add0n.com/useragent-switcher.html (参照 2023/11/27)