[Work in Progress] 研究報告

SPDYを利用したセキュアなセッション管理方式

鈴木 大作[1]　今泉 貴史[2]

SPDY-based secure session management system

1. はじめに

近年，多くの企業や組織がWebアプリケーションを提供している．しかし，Webアプリケーションの脆弱性を突いた攻撃は後を絶たず，度々深刻な被害が発生している．そうした攻撃の1つに，クライアント・サーバ間の連続した通信(セッション)が第三者に乗っ取られるセッションハイジャックがある．本研究では，Webアプリケーションのセキュリティを高める事を目的とし，SPDYを利用したセキュアなセッション管理方式を提案する．

2. 提案手法

本研究では，クライアント・サーバ間のTCPコネクションを長時間維持し，TCPコネクションと2種類のCookieを利用してクライアントを識別する**S3M(SPDY-based Secure Session Management)方式**を提案する．TCPコネクションを維持する手段として，本研究ではSPDY[1]というプロトコルに注目した．S3M方式では，コネクションに基づいて2種類のCookieを交換する事で，セッションIDが流出するリスクの低いセッション管理を実現する．S3M方式でのセッション管理の様子を図1に示す．

まず，クライアントAとの間にTCPコネクションC_Aを確立する．Webアプリケーションがクライアント AにCookie aを発行した場合，SPDYサーバはC_AとCookie aに対応する新しいCookie a'を発行し，Cookie aの代わりにクライアントAに送信する．これ以降SPDYサーバは，コネクションC_Aを通してa'が送られてきた場合には，対応するaに交換してWebアプリケーションへ渡す．もし攻撃者であるクライアントBがコネクションC_Bを通してa'を送信しても，コネクションC_Bとa'に対応するaが存在しないため，交換は起こらない．したがって，a'はそのままWebアプリケーションに送信されるが，Webアプリケーションはa'を発行していないため無効なCookieとして無視される．以上の方式により，結果としてセッションハイジャックを防ぐことができる．

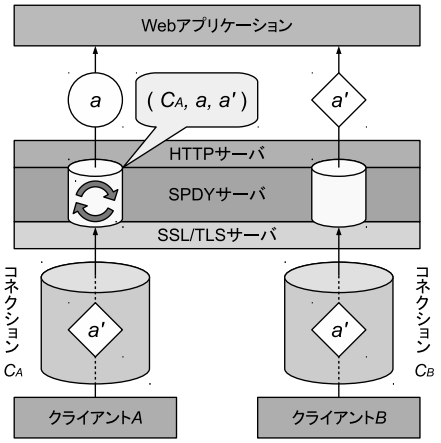

図 1　S3M方式の概要

3. 実装

本研究ではさらに，S3M方式を実現するSPDYサーバの実装を行った．S3M方式では，サーバと複数のクライアントの間でTCPコネクションを長時間接続しなければならない．そのため実装環境には，非同期で大量のアクセス処理を行う事が出来るNode.jsを採用した．Node.jsは，JavaScriptでサーバサイドを実装・実行するための環境である．今回はNode.jsでSPDYの機能を提供するモジュールであるnode-spdyを改良し，S3M方式を組み込んだシステム**secure-spdy**を作成した．

参考文献

[1] SPDY - The Chromium Projects. https://www.chromium.org/spdy.
[2] 独立行政法人 情報処理推進機構, セキュアプログラミング講座. http://www.ipa.go.jp/security/awareness/vendor/programmingv2/web.html. June. 2007

[1] 千葉大学大学院融合科学研究科
Graduate School of Advanced Integration Science, Chiba University
[2] 千葉大学統合情報センター
Institute of Management and Information Technologies, Chiba University