Computer Security Symposium 2017
23 - 25 October 2017

ブラウザストレージを利用した Web ブラウザに対する DoS 攻撃

上久保　遼[†1]　齊藤　泰一[†2]

概要：Web ブラウザにはストレージ環境(ブラウザストレージ)が用意されており、古くから Cookie があるが、HTML5 の登場とともにその機能は拡張され、その 1 つに WebStorage がある。WebStorage は主要ブラウザすべてで実装されており簡単に利用可能である。一方、ブラウザストレージを悪用したブラウザに対する DoS 攻撃が知られている。攻撃者がサイトの XSS 脆弱性を利用して寿命の長い巨大な Cookie をブラウザに設定することにより、ブラウザはそのサイトにアクセスできなくなってしまうという攻撃である。本稿では、ブラウザに対する DoS 攻撃として新たに WebStorage が悪用出来ることを示し、その機能と攻撃手法について報告する。

キーワード：DoS 攻撃，ブラウザ，WebStorage

DoS attack against the Web Browser using the browser storage

Ryo Kamikubo[†1]　Taiichi Saito[†2]

Abstract: Web browser has a storage function called cookie. A cookie is sent from a web server and stored in browser. WebStrage is another storage function included in HTML5. It is implemented in most browsers and can be easily used. On the other hand, a DoS attack method against web browser using the browser storage, cookie, are known. For example, attacker uses an XSS vulnerability of the site to set up a huge and longtime cookie in the browser, so that the browser can not access the site again. In this paper, we show that WebStorage can be exploited newly as a DoS attack against Web Browsers, and report on its function and attack method.

Keywords: DoS attack, Browser, WebStorage

1. はじめに

情報通信ネットワークが普及したことから DoS 攻撃(Denial of Service Attack)と呼ばれる攻撃対象のコンピュータに対して不正にリクエストを送るなどにより、リソースを意図的に消費しサービスを妨害する攻撃がある。これまでの DDoS 攻撃はおおかた下位レイヤ(Layer3/4)で行われていた。その攻撃手法は、踏み台となるクライアントをマルウェアに感染させてボットを形成し、そのボットに対して C&C サーバからコマンドを出すことによって攻撃を開始するというものである。一方で、上位レイヤ(Layer7)で行われる DDoS 攻撃が存在する。Layer7 で行われる攻撃手法は多種多様に渡り、その中の 1 つに web ブラウザを悪用した攻撃が存在する。これは通常の web ブラウザを踏み台として攻撃対象のコンピュータにリクエストを不正に送るという攻撃であるが、今回話題とする DoS 攻撃は、これまでの DoS 攻撃とは異なる web ブラウザを標的とした新しい攻撃手法である。web ブラウザを標的とした攻撃として知られているものにウェブストレージのひとつである cookie を悪用した攻撃例がある。

・ブラウザストレージ

Web ブラウザにはストレージ環境(ブラウザストレージ)が用意されており、古くから Cookie があるが、HTML5 の登場とともにその機能は拡張され、その 1 つに WebStorage がある。WebStorage は主要ブラウザすべてで実装されており簡単に利用可能である。一方、ブラウザストレージを悪用したブラウザに対する DoS 攻撃として Cookie を悪用したものが知られている。攻撃者がサイトの XSS 脆弱性を利用して寿命の長い巨大な Cookie をブラウザに設定することにより、ブラウザはそのサイトにアクセスできなくなってしまうという攻撃である。本稿では、ブラウザに対する DoS 攻撃として新たに WebStorage が悪用出来ることを示し、その機能と攻撃手法について報告する。

2. これまでに報告された攻撃

(1) Big Cookie

Big Cookie は 2012 年には既に確認され、未だに有効な攻撃手法である。その攻撃手法としては、巨大なサイズの cookie (Big Cookie)を不正に設定することによって起こる。JavaScript は 4KB を超える巨大なサイズの Cookie (Big Cookie)をブラウザに設定することが出来る。一方 HTTP サーバは受け取れる HTTP ヘッダーサイズの上限があり、それを超えていた場合に Bad Request を返す。Apache2 の場合

†1 東京電機大学大学院工学研究科情報通信工学専攻,〒120-8551 東京都足立区千住旭町５, Tokyo Adachi-ku Senjuasahi-cho 5, 16kmc08@ms.dendai.ac.jp

†2 東京電機大学，〒120-8551 東京都足立区千住旭町５, Tokyo Adachi-ku Senjuasahi-cho 5

デフォルトの HTTP ヘッダーサイズ上限は 8096KB である。XSS 脆弱性を利用し Big Cookie を設定されるとブラウザは以後サーバにアクセスしようとしても Bad Request となる為サービスが利用できなくなる。この攻撃ではブラウザ側から保存されている Cookie を削除することにより対策が可能であるが、もし Big Cookie の有効期限が非常に長く設定されていた場合、ユーザの操作なしに回復することは不可能である。以下に BigCookie 攻撃をした際のブラウザの動作を報告する。

実際の攻撃コードは JavaScript で以下の形で行った。

```
function bigcookie(){
document.cookie = '巨大なデータ';
}
```

図 2-1 は攻撃前の通常の web ページである。

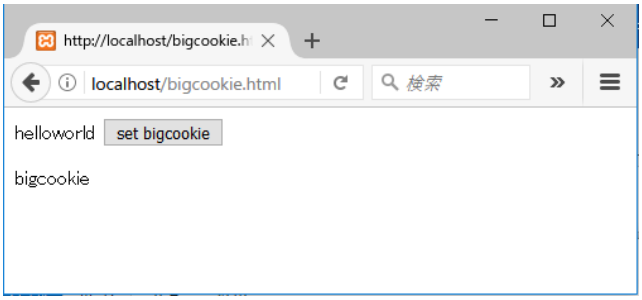

図 2-1　bigcookie 攻撃前のページ

Figre2-1 bigcookie attack before

図 2-2 は BigCookie 攻撃後の web ページである。

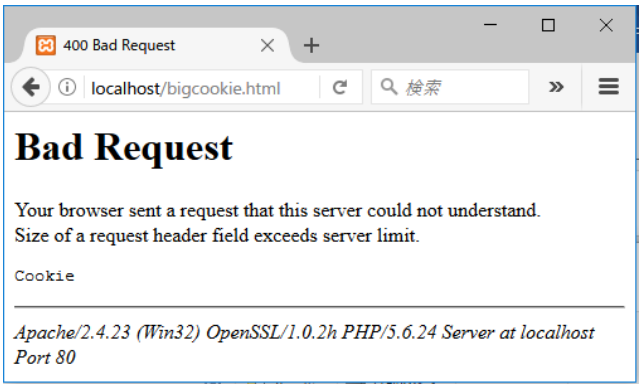

図 2-2　bigcookie 攻撃後のページ

Figre2-2 bigcookie attack after

図 2-3　セットされた巨大な cookie

figre2-3 set bigcookie

3. WebStorage

3.1 概要

WebStorage はブラウザがローカル環境にデータを保存するための API である。HTML5 の一部として定義され、現在は HTML5 から分離し独立した仕様として定義されている。最新の主要なブラウザすべてで機能の一部が互換性のある方法でサポートされている。WebStorage は基本的な機能は cookie と似ている。大きく異なる点として WebStorage は保存容量が大きい。WebStorage には sessionStorage と localStorage の 2 種類が用意されており、データの保存場所と有効期限が異なる。どちらもキーバリュー型のストレージであり、データの操作は JavaScript で簡単に行うことが出来る。以下に 2 種類の特徴を述べる。

3.2 sessionStorage

sessionStorage はセッション内のデータを保存する。データの保存期間は一時的で、作業中のタブやウィンドウに限定され、閉じるとデータは削除される。

(1) データの操作

key に対応するデータ(value)を保存するには次のように行う。key は数字以外にも利用可能である(例えば文字、文字列)。

```
sessionStorage.setItem(key,value);
```

key に対応するデータを取得するには次のように行う

```
var item = sessionStorage.getItem(key);
```

key に対応するデータの削除は以下のように行う。clear()を行うと sessionStorage 内のデータが全削除される。

```
ver item = sessionStorage/removeItem(key);
sessionStorage.clear();
```

3.3 localStorage

複数のセッションでデータを保存する場合に使用される。保存されたデータは操作がない限り永続する。基本的な操作は sessionStorage と同じ為省略する。

以下に Cookie 、sessionStorage、localStorage のデータの扱いをまとめる。

表 3-1　ウェブストレージによるデータの取り扱い

Table 3-1　Data handling by WebStorage.

	データの有効期限	データ量の上限
Cookie	指定期限まで有効	4KB
sessionStorage	ウィンドウ・タブを閉じるまで有効	5MB/origin
localStorage	永続的	5MB/origin

4. 攻撃

4-1 攻撃準備

sessionStorage を利用した場合の攻撃コードは以下である。localStorage の場合、sessionStorage を localStorage に書き換える。1 つのキーに対応するデータ(value)は 5MB のテキストデータを使用。

```
var storage = sessionStorage;
function set (){
var value = 大きいデータ;
for(i=1;i>0;i++){
storage.setItem(i,value);
}
```

4-2 実験環境

攻撃対象のブラウザを動作させるコンピュータは以下の環境である。

OS　　windows10Pro
CPU　　i7-6600U 2.5GHz*2
RAM　　PC3-12800 8*2GB
Storage　　SSD235GB

攻撃用 web ページは apache2.0 を用いてローカルに設置
実験ブラウザは以下である。
Firefox54.0.1 64bit
Google Chrome60.0.3112.101 64bit
//InternetExplorer11 11.540.15063.0

4-3 実験方法

攻撃用 web ページをブラウザから開き、その時の状態をタスクマネージャ及びリソースモニタで監視。各ブラウザにてそれぞれ実験を行う。

4-4 結果

・Firefox

sessionStorage、localStorage ともに攻撃時の動作はあまり差は出なかった。攻撃後メモリの使用率は実験環境のコンピュータが利用可能な上限 15.9GB まで張り付くことを確認した。また攻撃中はストレージの占有も確認した。攻撃開始の約 120 秒後からストレージの占有が始まり、ストレージの空き領域が約 50GB を切るとブラウザがクラッシュする動作を確認した。

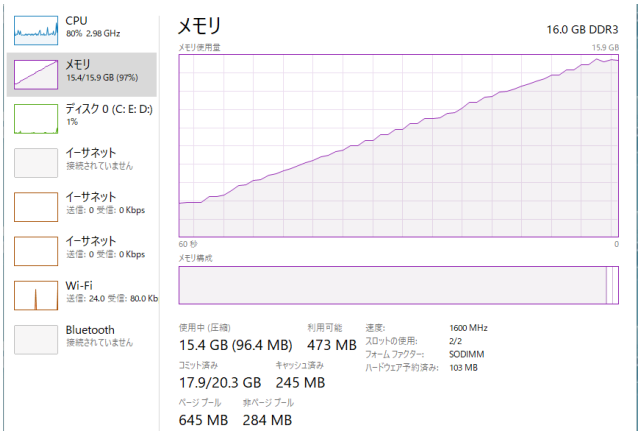

図 4-1　sessionStorage による攻撃時メモリ使用率(Firefox)

Figre4-1 Memory usage rate at attack by sessionStorage(Firefox)

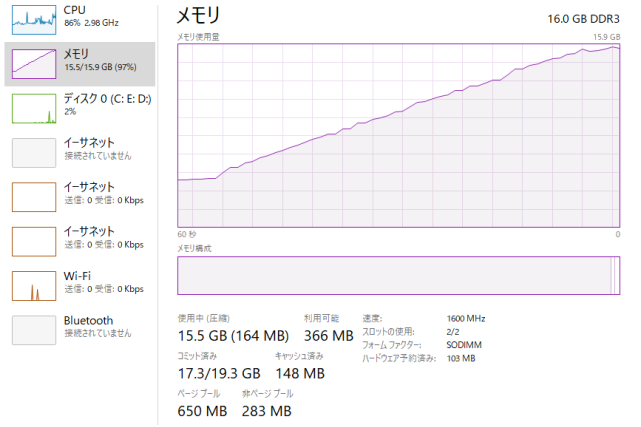

図 4-2　localStorage による攻撃時メモリ使用率(Firefox)

Figre4-2 Memory usage rate at attack by localStorage(Firefox)

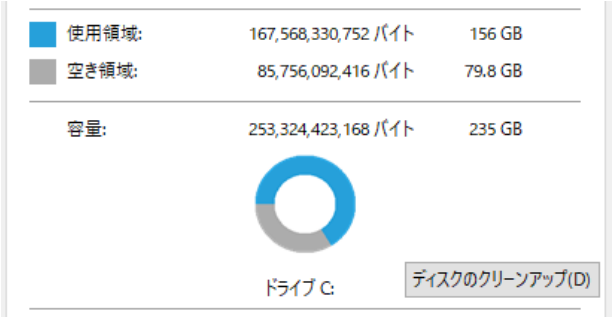

図 4-3　攻撃前のストレージの状態

Figre4-3 Storage status before attacks

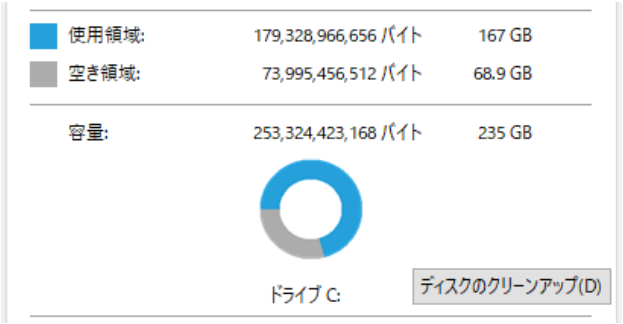

図 4-4　攻撃中のストレージの状態(Firefox)

Figre4-4 Storage status under attacking(Firefox)

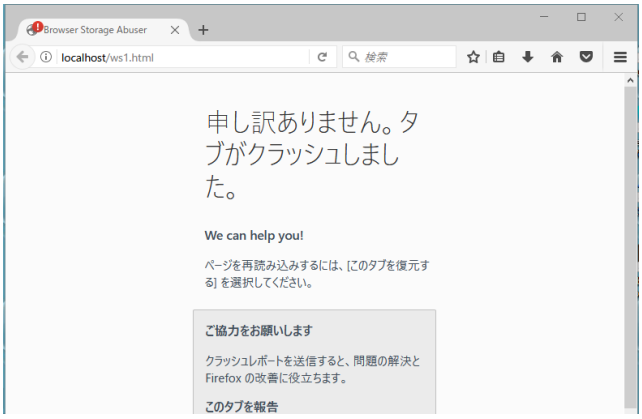

図 4-5　ブラウザのクラッシュ(Firefox)

Figre4-5 Browser crashed(Firefox)

・Chrome

Chrome でも Firefox 同様に sessionStorage と localStorage によるコンピュータに与える影響の差は殆ど無い。しかし Firefox とは異なり、sessionStorage、localStorage ともにメモリの使用率の増加はあまり見られなかった(図 4-6,4-7)。一方でストレージの占有は Firefox と比べて非常に高く、攻撃実行時から占有が始まり実行から 10 分程度で空き容量をすべて消費した(図 4-8)。しかし Firefox ではブラウザがクラッシュしたものの Chrome では著しく動作が遅くなるだけでクラッシュは発生しなかった。

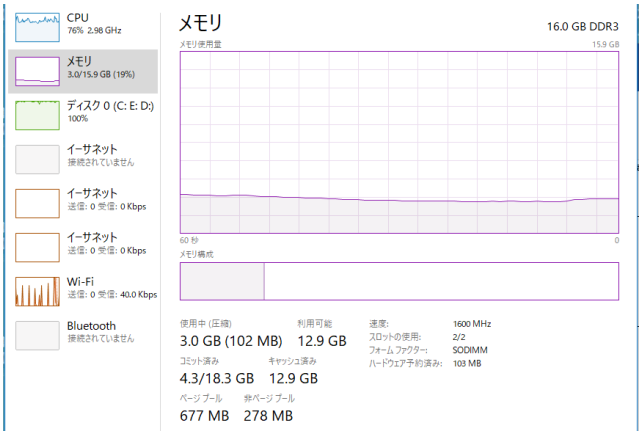

図 4-6　sessionStorage 攻撃時メモリ(Chrome)

Figre4-6 Memory usage rate at attack by sessionStorage(Chrome)

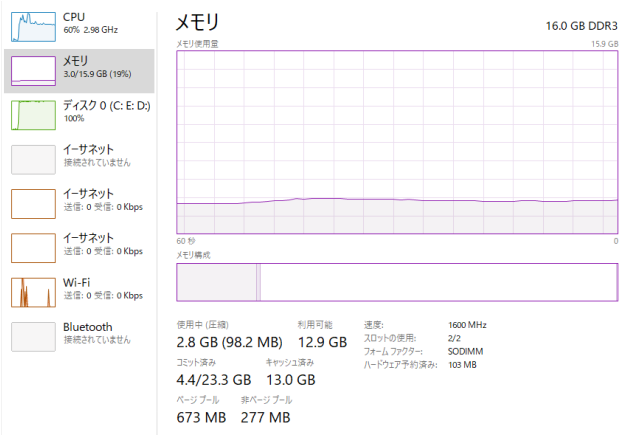

図 4-7　localStorage 攻撃時メモリ(Chrome)

Figre4-7 Memory usage rate at attack by localStorage(Chrome)

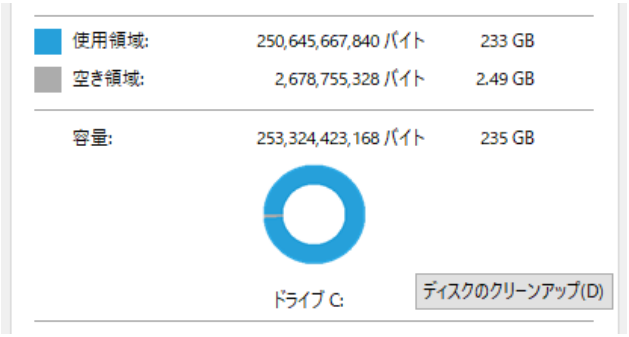

図 4-8　攻撃後のストレージの状態(Chrome)

Figre4-8 Storage status after attack(Chrome)

5.　評価

使用ブラウザに関わらず sessionStorage と localStorage を攻撃に利用した場合、動作に大きな違いはない。どちらを攻撃手法としてもメモリ及びストレージの使用率を不正に引き上げることが可能である。しかしその動作はブラウザによって異なる。

Firefox ではメモリがコンピュータの上限に到達してからストレージの占有が開始された。その為、今回の実験環境の場合攻撃開始から実際にストレージが占有されるまで 1 分ほどのタイムラグがあった。そして約 25GB 程度占有したところでブラウザのクラッシュを確認した。一方 Chrome ではメモリの使用率は上限に到達することはなかった。しかし攻撃開始時からストレージの占有が始まり、コンピュータの使用可能ストレージをほぼ全部埋め尽くすことが可能である。しかしブラウザのクラッシュは確認できなかった。ブラウザによる動作を表 5-1 にまとめる。

WebStorage を利用した DoS 攻撃の特徴として短時間で一時的に特定のブラウザを落とすことが可能である。今回

提案した WebStorage を利用した DoS 攻撃はブラウザベースの DoS 攻撃の防御法として利用可能ではないかと予測する。サーバ側がブラウザからの DoS 攻撃を検知した場合にブラウザに対して仕掛けることによって、効率的にかつ影響を最小限にブラウザを落とす、パフォーマンスを著しく低下させることによって攻撃そのものにアプローチ可能である。これについては今後の検討課題とする。

WebStorage は PC 向けブラウザだけではなくモバイル向けブラウザでも実装されている。今後の検討課題の 1 つとしてモバイル向けブラウザを含め今回実験していないブラウザ、例えば InternetExplorer や Edge などでも動作確認を行う必要がある。

表 5-1　WebStorage DoS 攻撃の評価
Table 5-1 Evaluation of WebStorage DoS attacks

	Firefox	Chrome
メモリ使用率	コンピュータが提供する上限まで占有	僅かに占有
ストレージ使用率	メモリが上限に到達し次第 占有途中でブラウザクラッシュが発生	コンピュータが提供する上限まで占有
ブラウザクラッシュ	○	×

参考文献

[1] David Flanagan, JavaScript 第 6 版, 2012, p.645-675
[2] " Web Storage－HTML5 の API、および、関連仕様". http://www.htmq.com/webstorage/, (参照 2017-08-24)
[3] " HTML Standard". https://html.spec.whatwg.org/multipage/webstorage.html, (参照 2017-08-24)
[4] " document.cookie - Web API インターフェイス | MDN". https://developer.mozilla.org/ja/docs/Web/API/Document/cookie, (参照 2017-08-24)
[5] " 400 Bad Request, Cookie Too Large - Chrome, IE, Firefox, Edge". http://www.thewindowsclub.com/400-bad-request, (参照 2017-08-24)
[6] " DOM ストレージの概要". https://msdn.microsoft.com/ja-jp/library/cc197062(v=vs.85).aspx, (参照 2017-08-24)
[7] "DOM Storage - DOM | MDN".https://developer.mozilla.org/ja/docs/DOM/Storage, (参照 2017-08-24)