OpenAM によるシングルサインオン可否判
定手順書

OSSTech 株式会社

更新日	2023 年 5 月 26 日
リビジョン	1.9

目次

1 要旨

本文書では、シングルサインオン対象となる Web アプリケーションが、OpenAM によりシングルサインオン可能であるか技術的に判定する方法を説明します。技術的に可能である場合でも、お客様のセキュリティルールや、運用ルールなどの諸処条件により適用できない場合もあることをご了承ください。

1.1 判定フロー

以下の図はシングルサインオン可否の判定フローです。それぞれの判定基準の詳細については後述します。

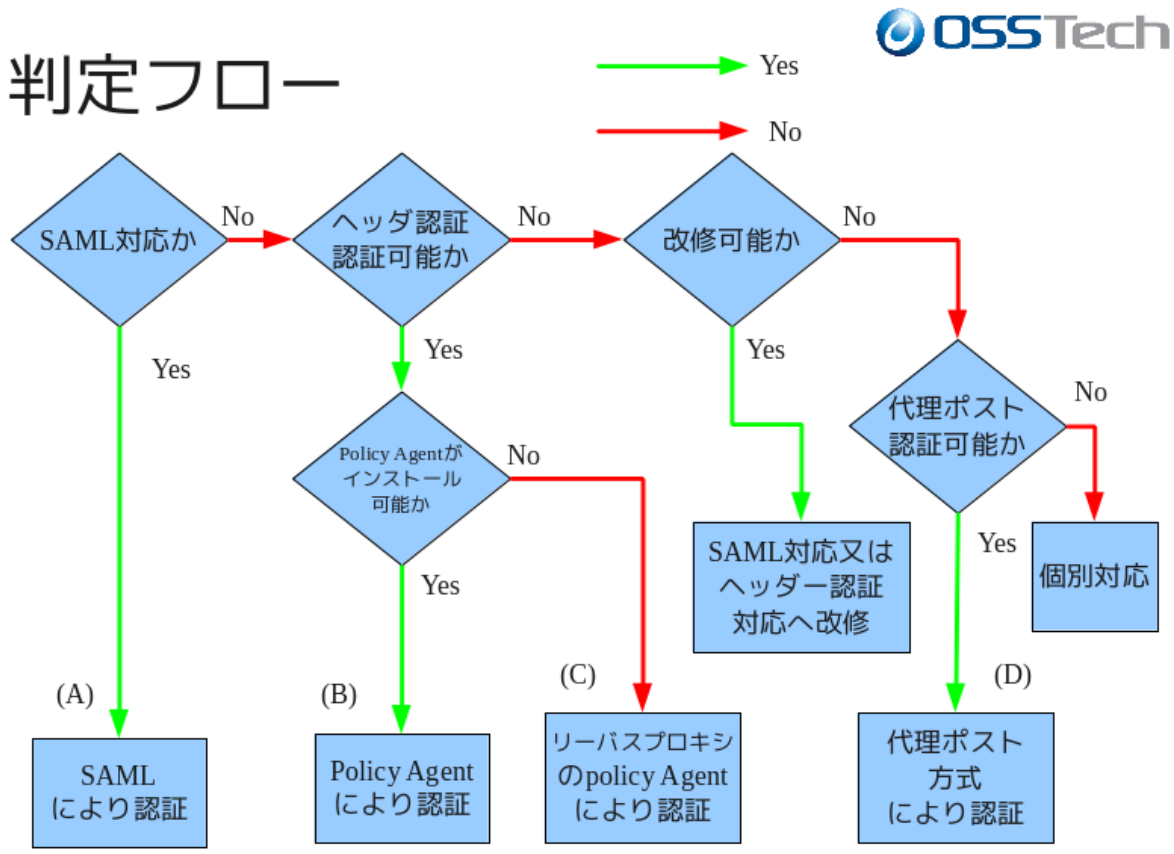

図 1　判定フロー

2 各シングルサインオン方式における可否条件

本章では、各シングルサインオン方式におけるシングルサインオン可否の判断基準について説明します。

2.1 SAML を利用したシングルサインオン

SAML を利用してシングルサインオンを実現するためには、以下の条件を満たす必要があります。

1. **シングルサインオン対象となる Web アプリケーション（以下 Web アプリケーション）が、SAML に準拠していること**

注意点シングルサインオン対象のアプリケーションが SAML 準拠の製品/ソフトウェアである場合、SAML メッセージに含まれる Extension 要素などの微調整が必要な場合があります。

この条件に当てはまらない場合は、以下に続く方式を検討します。

2.2 OpenAM Policy Agent を利用したシングルサインオン

OpenAM Policy Agent を利用してシングルサインオンを実現するために必要な条件について説明します。OpenAM Policy Agent を利用する場合、システム構成を以下の 2 つの構成から選択します。

1. **エージェント型**
2. **リバースプロキシ型**

2 つの構成で共通の条件と、各構成固有の条件を以下にまとめます。

2.2.1 共通条件

1. **Web アプリケーションが HTTP ヘッダー認証に対応していること**

HTTP ヘッダー認証とは、Web アプリケーションが HTTP リクエストに含まれるヘッダーの情報によりユーザーの本人確認を行い認証を行うような機能を指します。OpenAM の Policy Agent はユーザー情報を任意の名前の HTTP ヘッダーに設定して、Web アプリケーションに渡すことが可能です。

OpenAM Policy Agent を利用してシングルサインオンを実現する場合は、Web アプリケーションが HTTP ヘッダー認証に対応している必要があります。

2.2.2 エージェント型シングルサインオンが可能な条件

「共通条件」に加えて、以下の条件を満たす必要があります。

1. Web アプリケーションが動作する Web サーバ、コンテナに OpenAM Policy Agent がインストール可能であること

この条件に当てはまらない場合は、以下に続く方式を検討するか、Web アプリケーションの改修を検討します。

2.2.3 リバースプロキシ型シングルサインオンが可能な条件

「共通条件」に加えて、以下の全ての条件を満たす必要があります。

1. リバースプロキシを導入可能であること。
2. リバースプロキシとして動作する Web サーバーに OpenAM Policy Agent がインストール可能であること。

この条件に当てはまらない場合は、以下に続く方式を検討するか、Web アプリケーションの改修を検討します。

2.3 代理 HTTP POST 型シングルサインオン

代理 HTTP POST 型とは、ユーザーの代わりにプログラムに ID/PW を送信させる方式です。アプリケーションの改修が不可能な場合に、この方式を採用します。

代理 HTTP POST 方式を利用してシングルサインオンを実現するためには、以下の全ての条件を満たす必要があります。

1. 他のコンテンツには無いログイン画面だけに含まれる文字列がある
または、ログイン画面に任意の文字列を埋め込める
2. 送信する認証情報の input タグの name 属性が固定である
3. ログインするために送信が必要な認証情報は「OpenAM データストア」
「ログイン画面 input タグ内」「設定ファイル記載固定文字列」のいづれかである。
4. ログイン操作の実行に JavaScript を必要としない
5. 認証が成功する場合と同じ HTTP リクエストを再現することで再度認証が可能である

代理認証の詳細は「OpenAM 代理認証オプション (mod_authproxy) 管理者ガイド」を参

照してください。

この条件に当てはまらない場合は、個別に調査するか、Web アプリケーションの改修を検討します。

2.4 Shibboleth を利用したシングルサインオン

Shibboleth を利用してシングルサインオンを実現するためには、以下のいずれかの条件を満たす必要があります。

a) Web アプリケーションが学術認証フェデレーションのサービスプロバイダー（SP）に登録されている。

b) Web アプリケーションが米国 Internet2(※) の Shibboleth Service Provider を利用している

c) Web アプリケーションが米国 Internet2(※) の Shibboleth Identity Provider と連携しシングルサインオンを実現した実績がある

学術認証フェデレーションのサービスプロバイダーについては以下の URL をご参照ください。http://www.gakunin.jp/docs/fed/participants

これらの条件に当てはまらない場合は、Web アプリケーションを個別に調査する必要がございます。

※米国 Internet2 が開発した Shibboleth は http://shibboleth.net/から入手可能です。

3 各種調査手順

3.1 HTTP リクエスト再現調査

本章では、代理 HTTP POST 方式で Web アプリケーションが認証可能であるか判定する手順を説明します。Chrome の開発ツール (デベロッパーツール) を使用して、Web アプリケーションのログイン機能の解析を行います。

3.1.1 デベロッパーツールの使い方

デベロッパーツールの使い方を説明します。「メニュー」(アドレスバーの右にある、三本線のボタン) - 「その他のツール」 - 「デベロッパーツール」[*1]からデベロッパーツールを開始します。

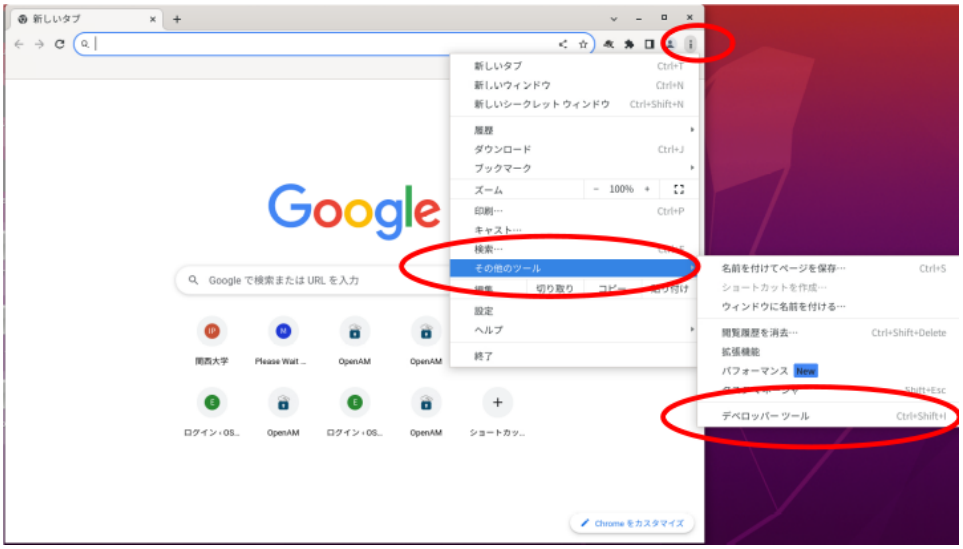

図 2　デベロッパーツールの開始

[*1] [ctrl] + [Shift] + [i] の同時押しでも起動できます。

「Network」タブを押します。

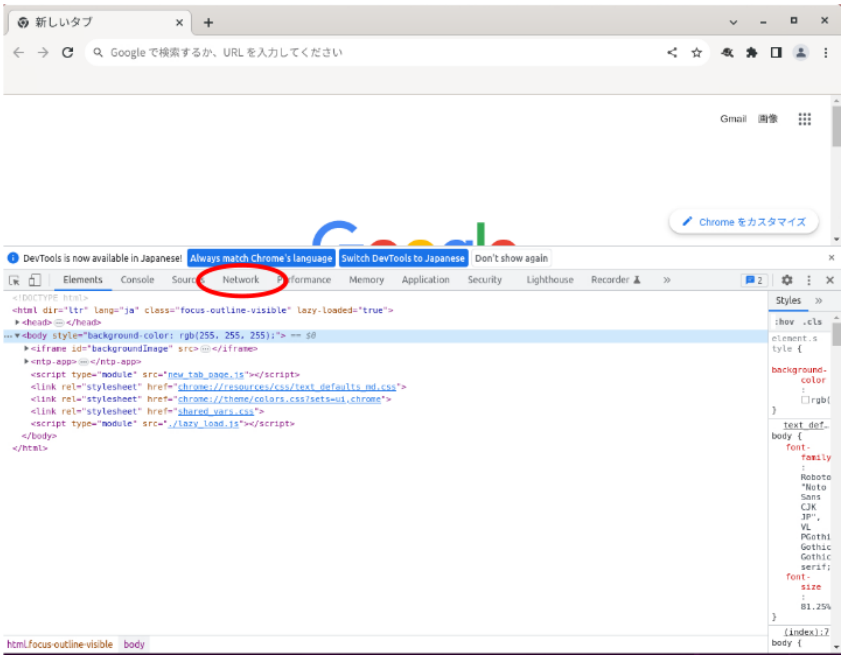

図 3　デベロッパーツールの elements タブ

「Preserve log」と「Disable cache」にチェックを入れます。

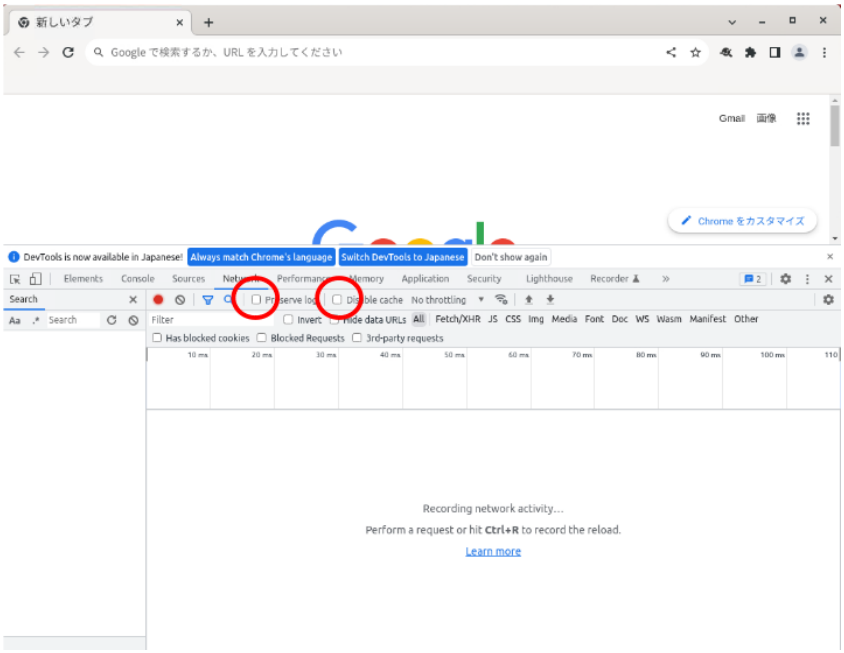

図 4　デベロッパーツールの Network タブ

「Preserve log」と「Disable cache」にチェックが入っていることを確認します。 recording のボタンが赤い状態で Web サイトにアクセスするとキャプチャーされます。ボタンが灰色の場合は、クリックして recording を開始 (赤い状態) にします。試しに google 検索で test と入力し検索します。

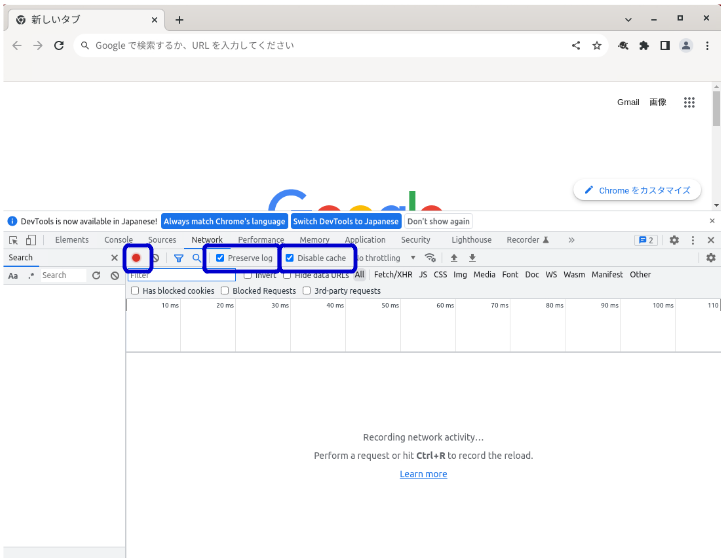

図 5　デベロッパーツールの Network タブ

Network タブの情報が更新され、HTTP のやりとりがキャプチャーされます。Clear を押すことでキャプチャーデータを削除出来ます。

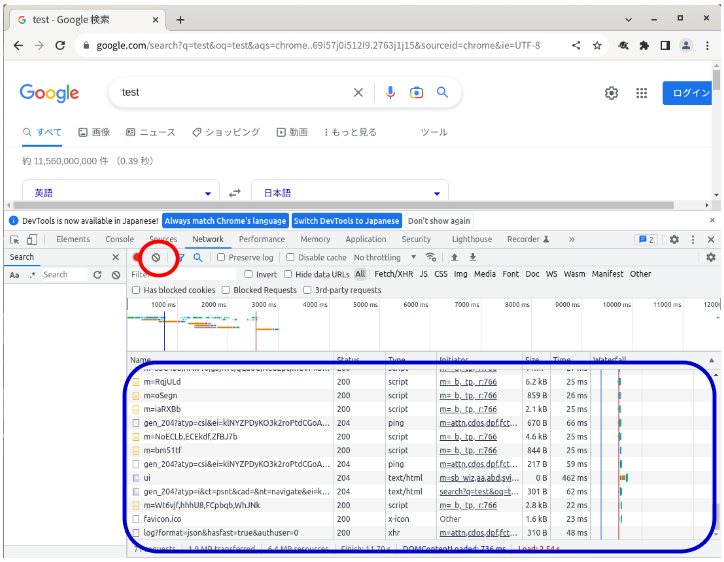

図 6　デベロッパーツールの Network タブ

キャプチャーデータが削除されました。

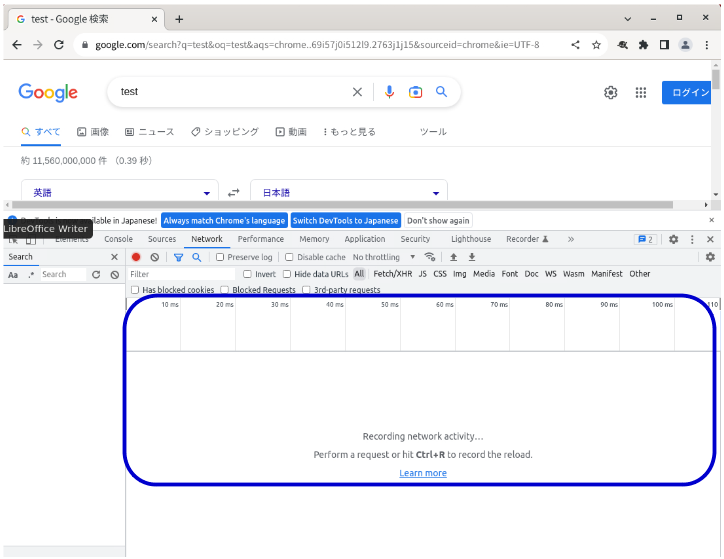

図 7　デベロッパーツールの Network タブ

3.1.1.1 【参考】新しいタブで開かれるページの情報取得

本内容は参考情報です。エラー調査等の目的で HTTP リクエストをキャプチャーする際、Web ページが新しいタブで開かれるとデベロッパーツールが起動しておらず HTTP リクエストの情報取得が出来ません。新しいタブを開いた際も情報を取得するデベロッパーツールの設定方法を説明します。

デベロッパーツールの設定 (歯車のアイコン) をクリックします。

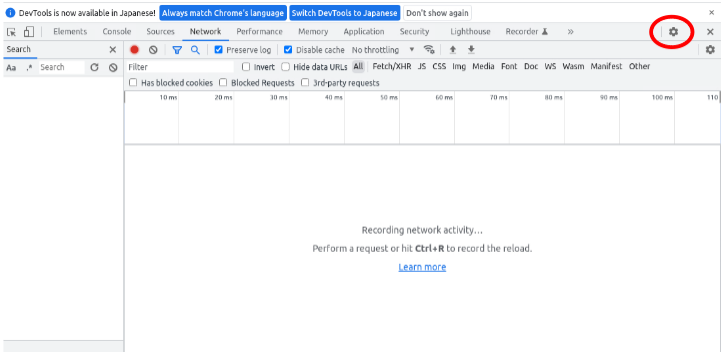

図 8　デベロッパーツールの設定

「Preferences」の「Global」の「Auto-open DevTools for popups」にチェックを入れます。

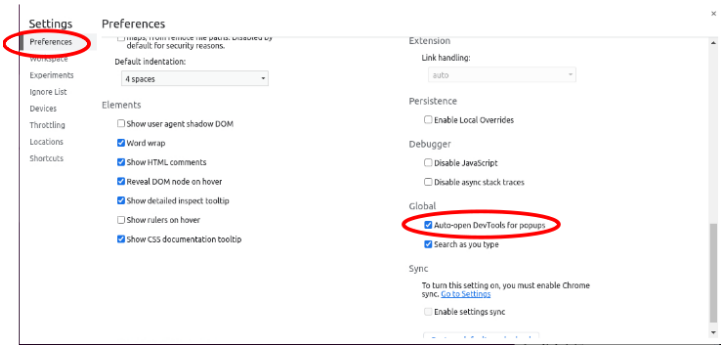

図 9　「Auto-open DevTools for popups」にチェック

「Auto-open DevTools for popups」にチェックが入っていると、新しいタブが開いた際にデベロッパーツールが起動した状態となり、新しいタブ上で HTTP リクエストの情報を取得することが出来ます。

3.1.1.2 【参考】HAR ファイルの取得

本内容は参考情報です。 HTTP リクエストをキャプチャーしたファイルを HAR ファイルとして保存 (export) する方法を説明します。HAR ファイルはエラー調査の目的でサポート担当より取得をお願いする場合があります。

デベロッパーツールの Netwoks を開いた状態で調査対象の事象を発生させ、HTTP リクエストを記録します。

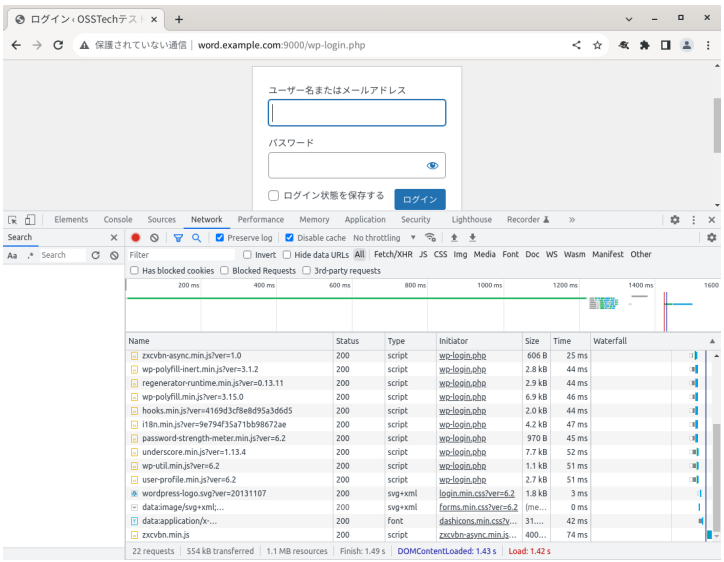

図 10　HTTP リクエストを記録

HAR ファイルの Export ボタンを押します。

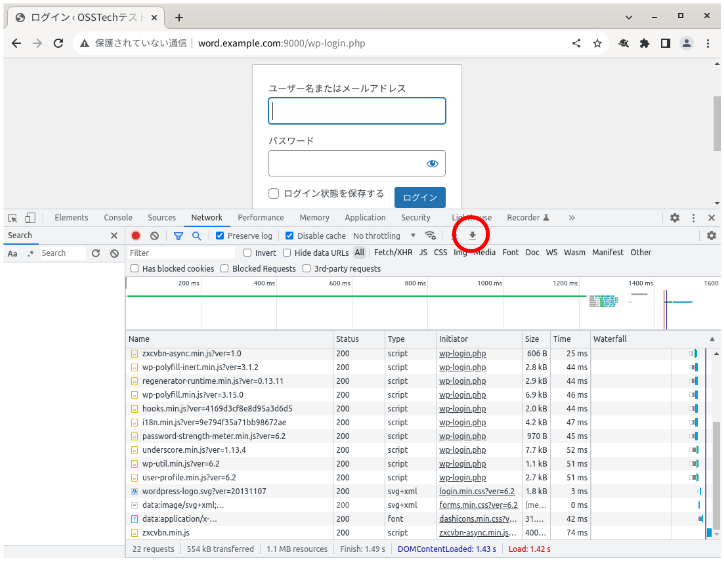

図 11　HAR ファイルの Export ボタンを押下

HAR ファイルを任意の場所に保存します。

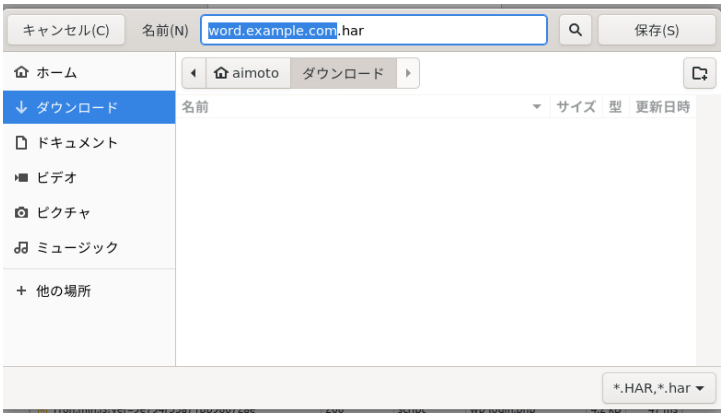

図 12　HAR ファイルの保存

以上で HAR ファイルの取得は完了です。HAR ファイルはテキストファイル (json データ) です。ID やパスワードなどを入力していた場合は、そのままファイルに含まれています。HAR ファイルに含まれるマスクすべき情報は、テキストエディタや sed コマンドなどで置換して下さい。

3.1.2 HTTP リクエストの再現

ログインのテストに備えて、ログイン済みとなっている Web アプリケーションからログアウトし、ログイン画面を再度表示させます。

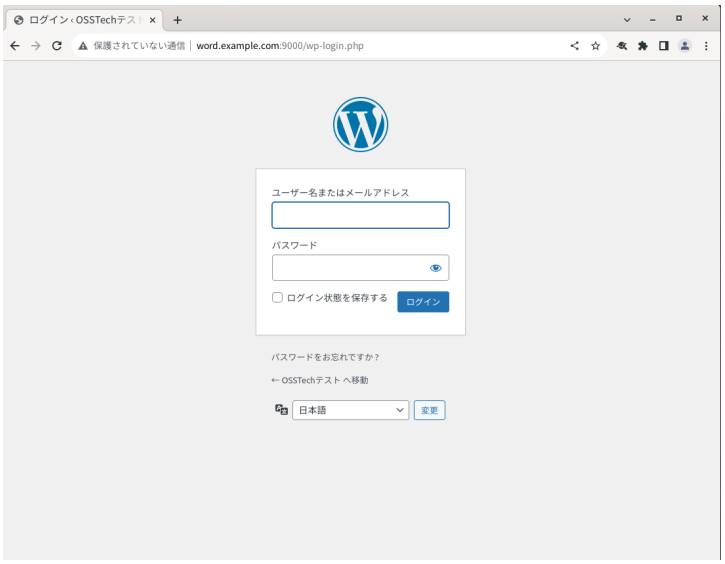

図 13　ログイン画面

デベロッパーツールを開始します。

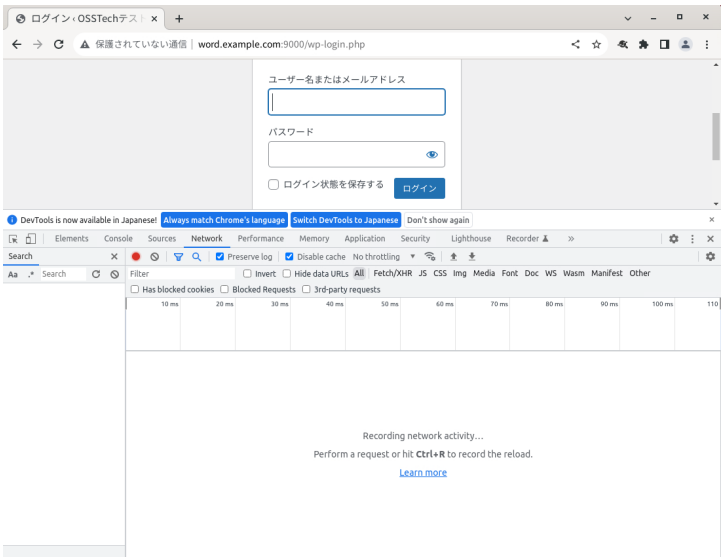

図 14　デベロッパーツールの開始

ログイン画面に ID/パスワードを入力してログインを行います。

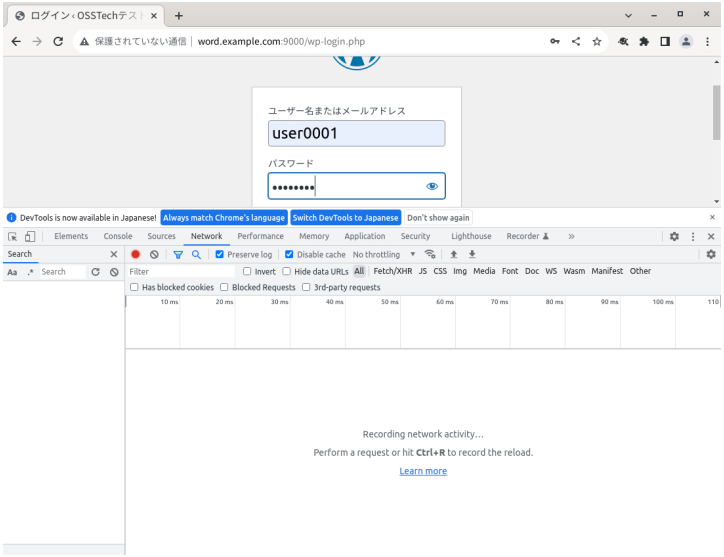

図 15　アプリケーションにログインを実施

アプリケーションのログインに成功することを確認します。

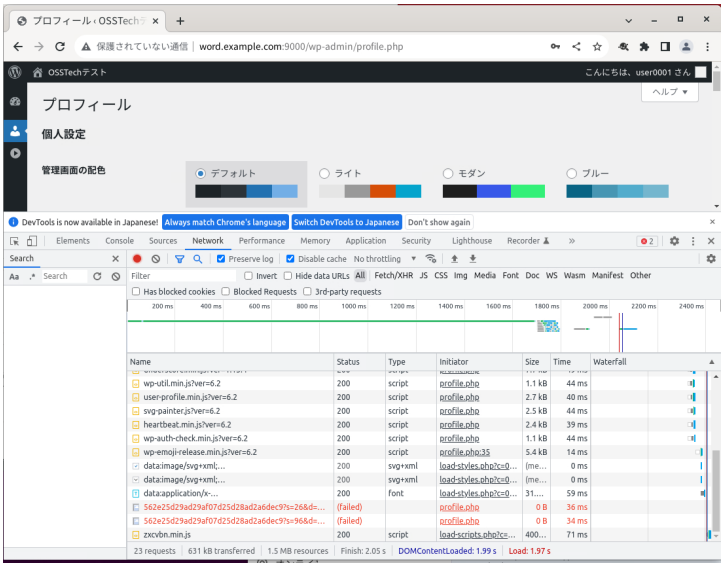

図 16　アプリケーションにログイン成功

デベロッパーツールの記録からログインリクエストを探します。ログイン画面表示から記録をスタートしていれば一番最初の記録がログインリクエストです。

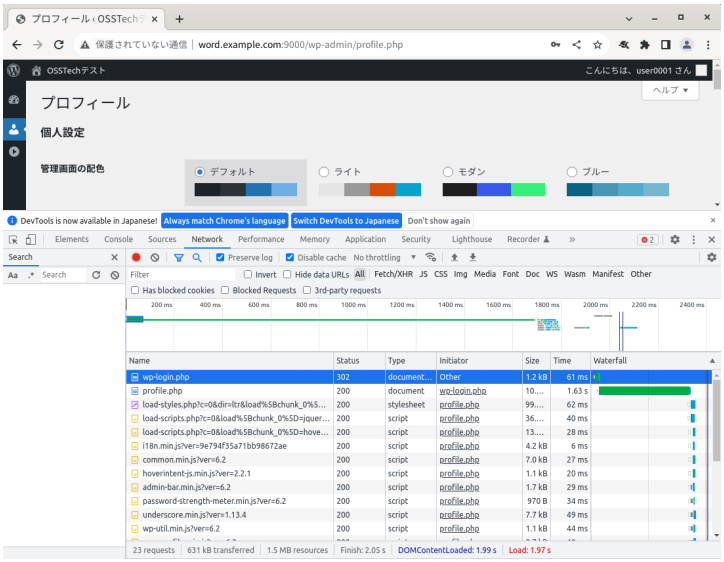

図 17　ログインリクエストの調査

ログインリクエストを押すとリクエストの詳細が表示されます。「Headers」タブより Request Method が POST であることを確認します。

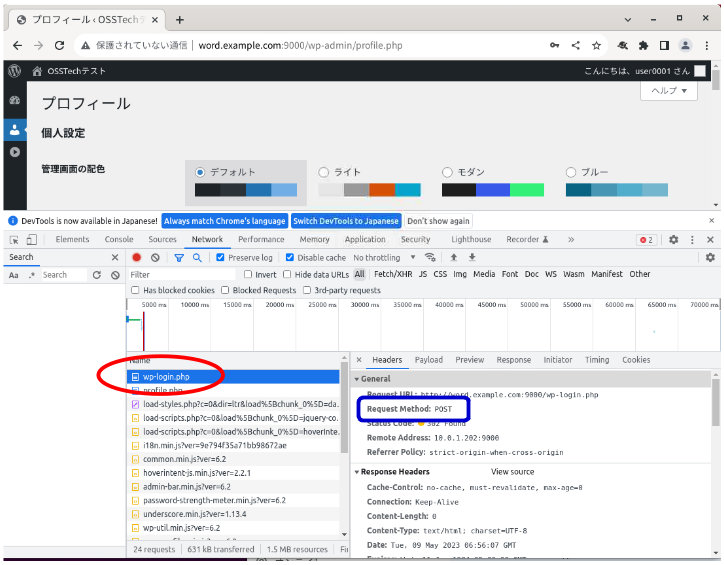

図 18　ログインリクエストの確認

「Payload」タブでログイン実行時にアプリケーションに送付した POST データを参照できます。ログイン画面で入力した ID やパスワードが入っていればログイン時に送信したデータです。例として下記の画像の場合は 5 つの POST データを送付していることが確認出来ます。

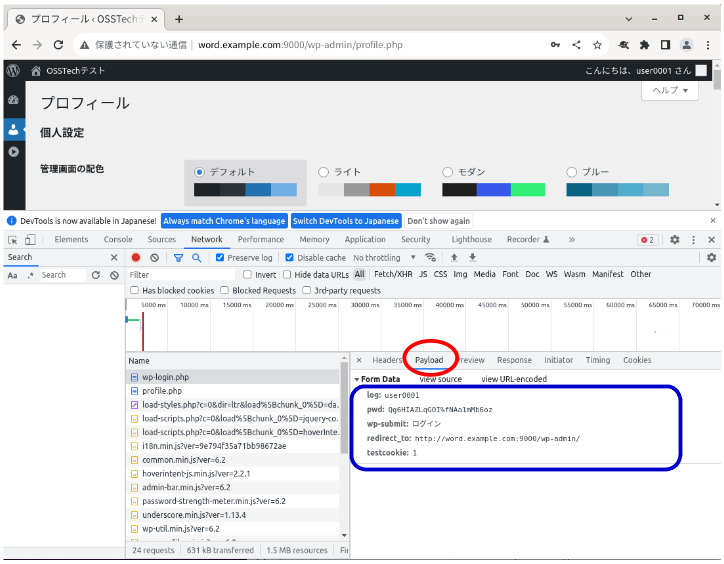

図 19　POST データの確認

送信しているデータが代理認証で用意できるかを確認します。代理認証で用意できるの値は「OpenAM データストアの値 (リクエスト HTTP ヘッダーの値)」「ログイン画面 input タグ内」「設定ファイルに記載した固定の値」です。 ログイン画面 input タグ内は後述の HTML の解析で確認します。POST データが全て用意できれば代理認証は可能と考えられます。

POST リクエストを再現して送信し、アプリケーションにログイン出来ればより正確に代理認証可能と確認出来ます。curl コマンドを用いて POST リクエストを再現して送信する手順例を示します。ログインリクエストを右クリックし「Copy as cURL」を選択して curl コマンドを取得します。

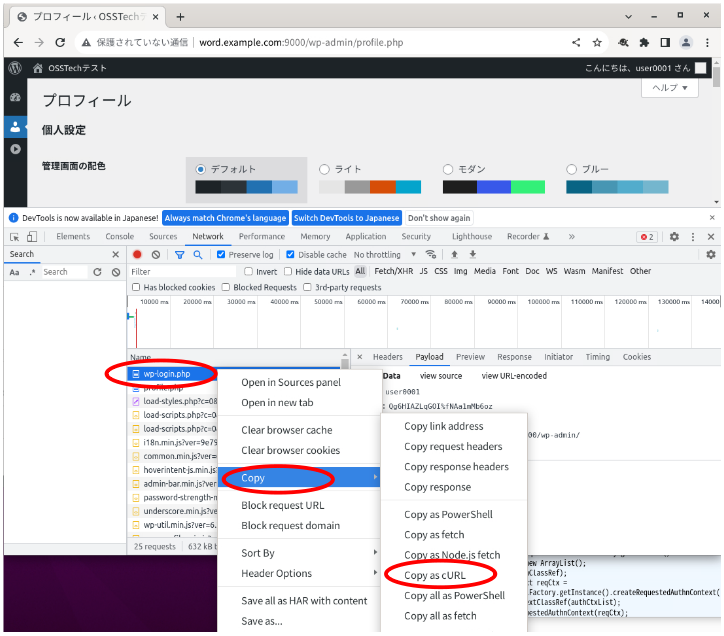

図 20　curl としてコピー

コピーした curl コマンドを Linux 端末から実行します。

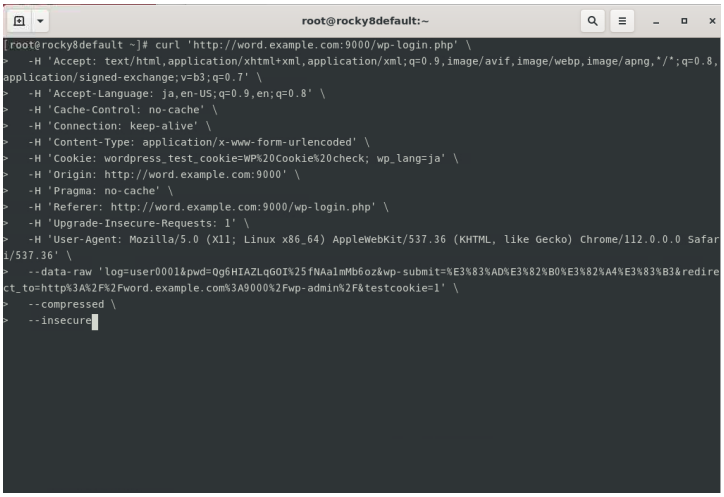

図 21　curl として実行

curl コマンド実行後にアプリケーションのログを確認し、ログインに成功していれば代理

認証可能です。ログインに失敗している場合は、「ログイン画面 input タグ内」の値が CSRF 対策で毎回変わる可能性が考えられます。事前に curl コマンドでログイン画面を表示し、POST するデータの値を取得するなどし調整して試してください。

3.2 HTML 解析

本章では Chrome の開発ツールを使用して、ログインページの HTML を解析する方法を説明します。

3.2.0.1 HTML の解析

1. **アプリケーションのログイン画面を表示します。**

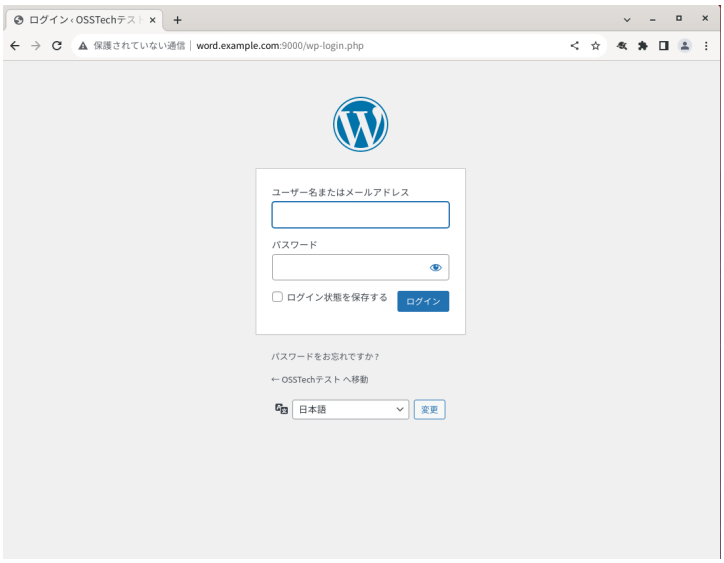

図 22　ログイン画面

2. デベロッパーツールを起動し Elements タブを表示します。

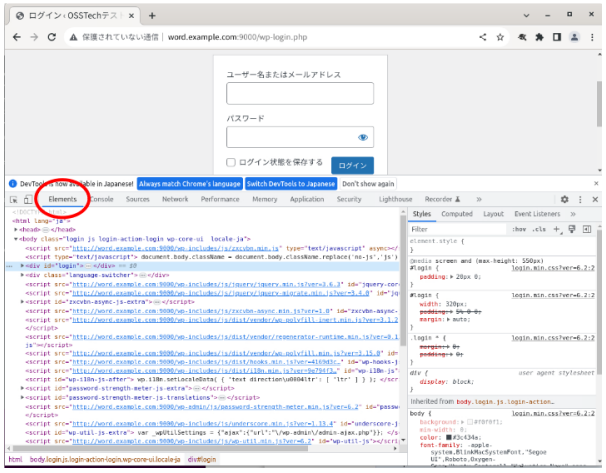

図 23　デベロッパーツールを起動

3. ユーザ ID の name 属性を確認します。

ユーザ ID に該当する入力項目に対応した HTML の input タグを探します。name 属性にランダムな数字が付加されていない事や、ブラウザの更新により変化しないことを確認してください。

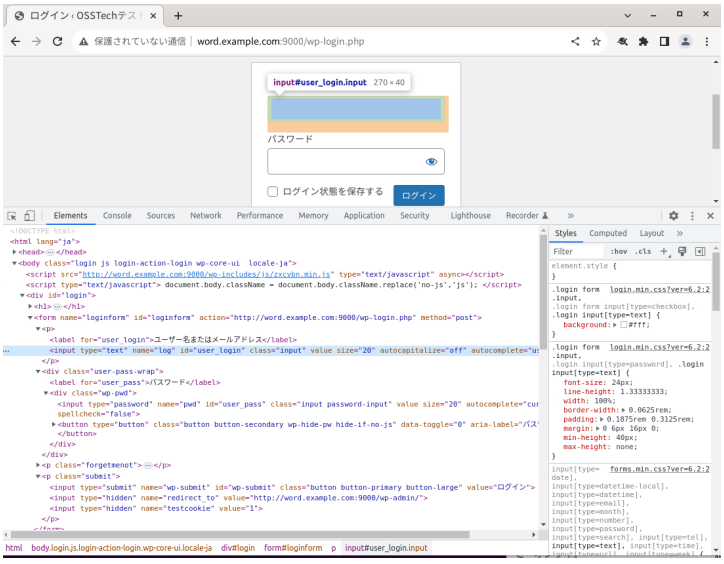

図 24　HTML 解析 1

4．パスワードの name 属性を確認します

パスワード入力項目に対応した HTML の input タグを探します。name 属性にランダムな数字が付加されていない事や、ブラウザの更新により変化しないことを確認してください。

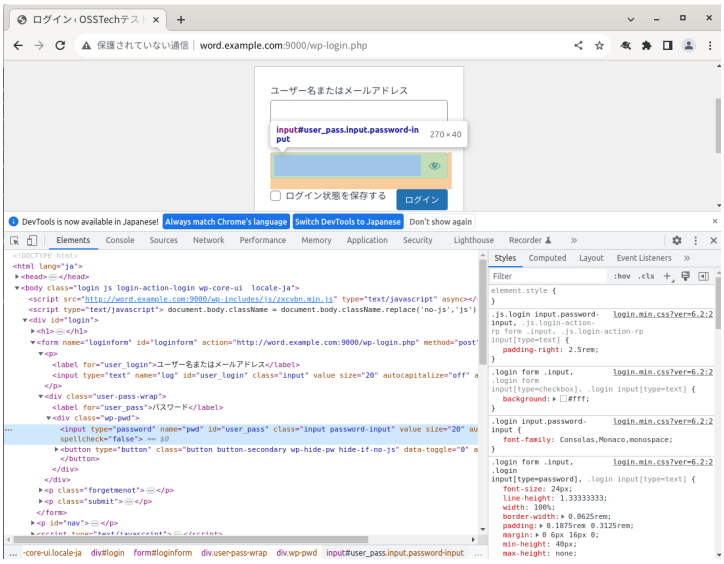

図 25　HTML 解析 2

5. パスワード以外に送付している値（input タグ）の name 属性を確認します

HTML の input タグを探します。name 属性にランダムな数字が付加されていない事や、ブラウザの更新により変化しないことを確認してください。

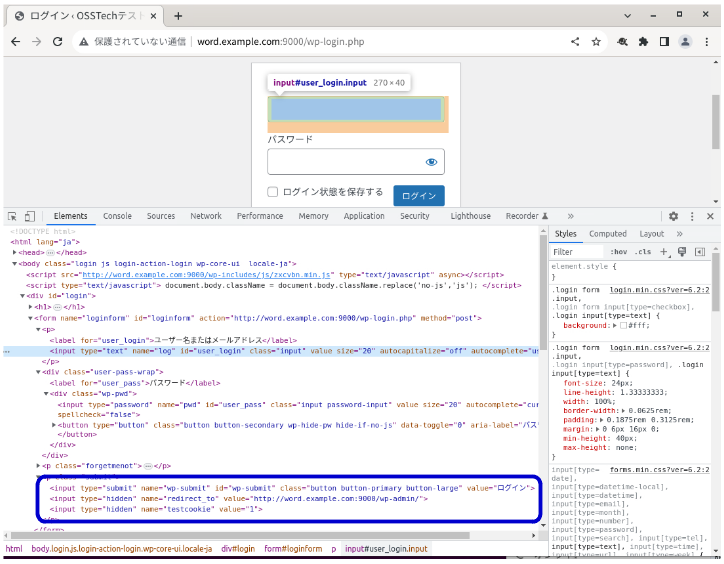

図 26　HTML 解析 3

6. ログイン処理に JavaScript が使用されていないか確認します

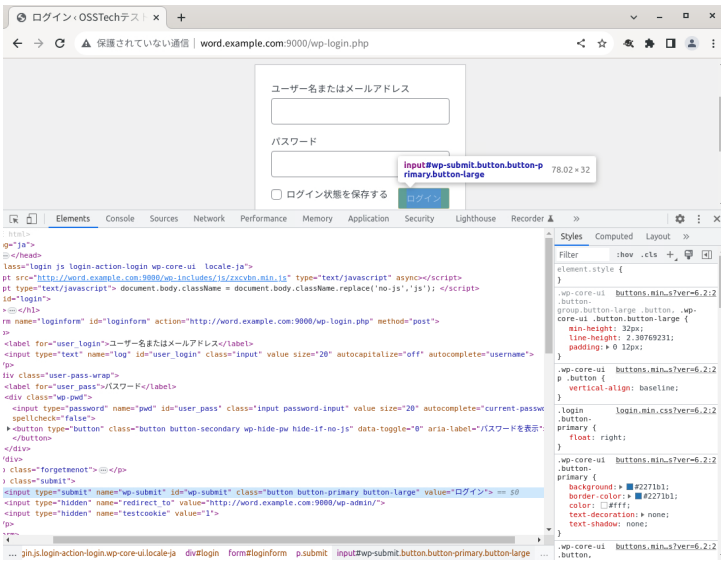

図 27　HTML 解析 4

ログインボタンを確認してください。JavaScript が設定されていないことを確認します。下記は JavaScript が設定されている例です。

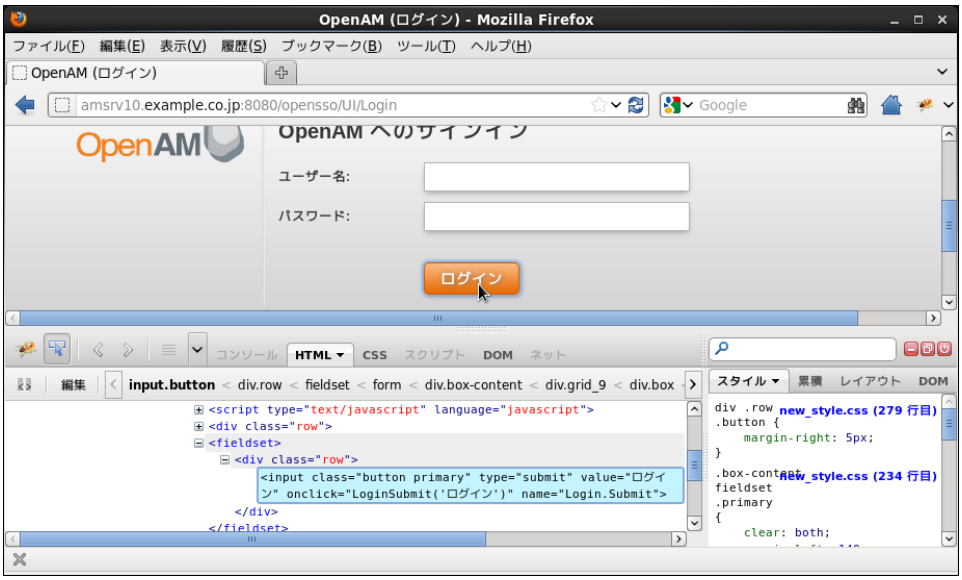

図 28　JavaScript の例

6．ログインフォームの名称を確認します

ツールバーの部分で「form」を選択してください。表示される fome タグの name 属性を確認します。

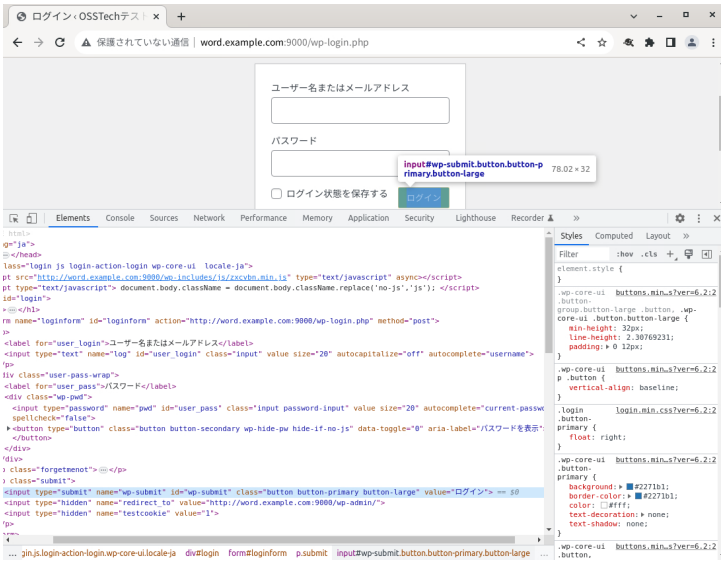

図 29　HTML 解析 4

4 改版履歴

- 2011 年 9 月 7 日 リビジョン 1.0
  - 初版作成
- 2012 年 5 月 22 日 リビジョン 1.2
  - 代理 HTTP POST 型シングルサインオンの条件を追加
- 2012 年 5 月 22 日 リビジョン 1.3
  - 章構成などを変更
  -「2.4Shibboleth を利用したシングルサインオン」を追記。
- 2012 年 7 月 3 日 リビジョン 1.4
  -「ポスト」を「HTTP POST」に置換
- 2012 年 11 月 13 日 リビジョン 1.5
  -「2.3 代理 HTTP POST 型シングルサインオン」を修正
- 2015 年 1 月 7 日 リビジョン 1.6
  - 表紙及び改版履歴を修正
- 2017 年 5 月 11 日 リビジョン 1.7
  - ドキュメント形式を Markdown に変更
- 2022 年 7 月 14 日 リビジョン 1.8
  - 表紙の社名を OSSTech 株式会社に変更
- 2023 年 5 月 26 日 リビジョン 1.9
  - HTTP リクエストの調査方法を Chrome のデベロッパーツールで実施する手順に変更
  - 参考情報として「新しいタブで開かれるページの情報取得」「HAR ファイルの取得」を追加