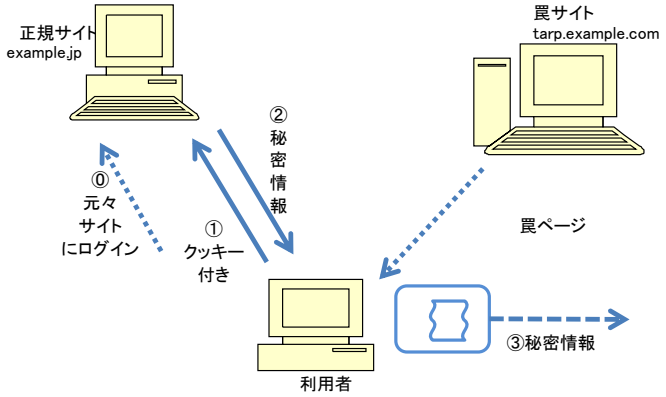
4.12.4 PDFのFormCalcによるコンテンツハイジャック(IEでアクセス)

罠ページのFormCalcがHTTPリクエストを呼び出し、結果を受け取る



PDFはFormCalcというスクリプト言語が使用でき、PDFドキュメントにFormCalcスクリプトを埋め込むことができます
FormCalcのURL関数によって、HTTPのやり取りの結果、秘密情報を受け取ることができる脆弱性があります
PDFのFormClacスクリプトから正規サーバにHTTPリクエストを送信するときに、クッキーも送信されるため、
JavaScriptとFormCalcの連携によって、秘密情報が漏洩されてしまいます

31-020 :ログイン
コンテンツハイジャック(PDF)

【ブラウザ】

- 4.12.4 PDFのFormCalcによるコンテンツハイジャック(IEでアクセス)
  1. 31-020 :ログイン ①
  2. コンテンツハイジャック(PDF) ②
  3. 4c-011d:ファイルアップロード（FormCalc対策版）
  4. コンテンツハイジャック(PDF; 対策版)

以下はコンテンツハイジャック攻撃用データ。

1. ContentHijacking.pdf

phpinfo
ホームに戻る

```
<li>4.12.4 PDFのFormCalcによるコンテンツハイジャック(IEでアクセス)</li>
<ol>
<li><a href="31/31-020.php">31-020 :ログイン</a></li>
<li><a href="http://trap.example.com/4c/ContentHijackingLoader.html?objfile=http://example.jp/4c/4c-
011a.php%3ffile%3dContentHijacking.pdf&objtype=pdf&target=http://example.jp/31/31-022.php&postdata=&logmode=all&regex=">コンテンツハイジャック(PDF)</a></li>
<li><a href="4c-011d.php">4c-011d:ファイルアップロード（FormCalc対策版）</a></li>
<li><a href="http://trap.example.com/4c/ContentHijackingLoader.html?objfile=http://api.example.net/4c/4c-
011d.php%3ffile%3dContentHijacking.pdf&objtype=pdf&target=http://example.jp/31/31-022.php&postdata=&logmode=all&regex=">コンテンツハイジャック(PDF; 対策版)</a></li>
</ol>
以下はコンテンツハイジャック攻撃用データ。
<ol>
<li><a href="ContentHijacking.pdf" download="ContentHijacking.pdf">ContentHijacking.pdf</a></li>
</ol>
</ul>
<a href="phpinfo.php">phpinfo</a><br>
<a href="/">ホームに戻る</a>
</body>
</html>
```

【サーバ： 31/31-020.php 】

/var/www/html/31/31-020.php - wasbook@example.jp - エディタ - WinSC

```
<?php
  session_start();  // セッションの開始
?>
<html>
<head><title>ログインしてください</title></head>
<body>
<form action="31-021.php" method="POST">
ユーザ名<input type="TEXT" name="ID"><BR>
パスワード<input type="PASSWORD" name="PWD"><BR>
<input type=SUBMIT value="ログイン">
</form>
</body>
</html>
```

【サーバ： 31/31-021.php 】

/var/www/html/31/31-021.php - wasbook@example.jp - エディタ - WinSCP

```
<?php
  session_start();  // セッションの開始
  $id = $_POST['ID'];
  $pwd = $_POST['PWD'];
  // IDとパスワードのどちらかが空の場合はログイン失敗
  if ($id == '' || $pwd == '') {
    die('ログイン失敗');
  }
  $_SESSION['ID'] = $id;
?>
<html>
<head><title>ログイン</title></head>
<body>
ログイン成功しました
<a href="31-022.php">プロフィール</a>
</body>
</html>
```

【サーバ： 31/31-022.php 】

/var/www/html/31/31-022.php - wasbook@example.jp - エディタ - WinSCP

```
<?php
  session_start();  // セッションの開始
  $id = $_SESSION['ID'];
  if ($id == '') {
    die('ログインしてください');
  }
?>
<html>
<head><title>プロフィール</title></head>
<body>
ユーザID:<?php echo htmlspecialchars($id, ENT_NOQUOTES, 'UTF-8'); ?>
</body>
</html>
```

【サーバ： 4c/4c-013a.php 】

/var/www/html/4c/4c-013a.php - wasbook@example.jp - エディタ - WinSCP

```
<?php
define('UPLOADPATH', '/var/upload');
$mimes = array('pdf' => 'application/pdf');

$file = $_GET['file'];
$info = pathinfo($file);       // ファイル情報の取得
$ext = strtolower($info['extension']);     // 拡張子
$content_type = $mimes[$ext]; // Content-Typeの取得
if (! $content_type) {
  die('拡張子はpdfを指定ください');
}
header('Content-Type: ' . $content_type);
header('X-Content-Type-Options: nosniff');
readfile(UPLOADPATH . '/' . basename($file));
?>
```

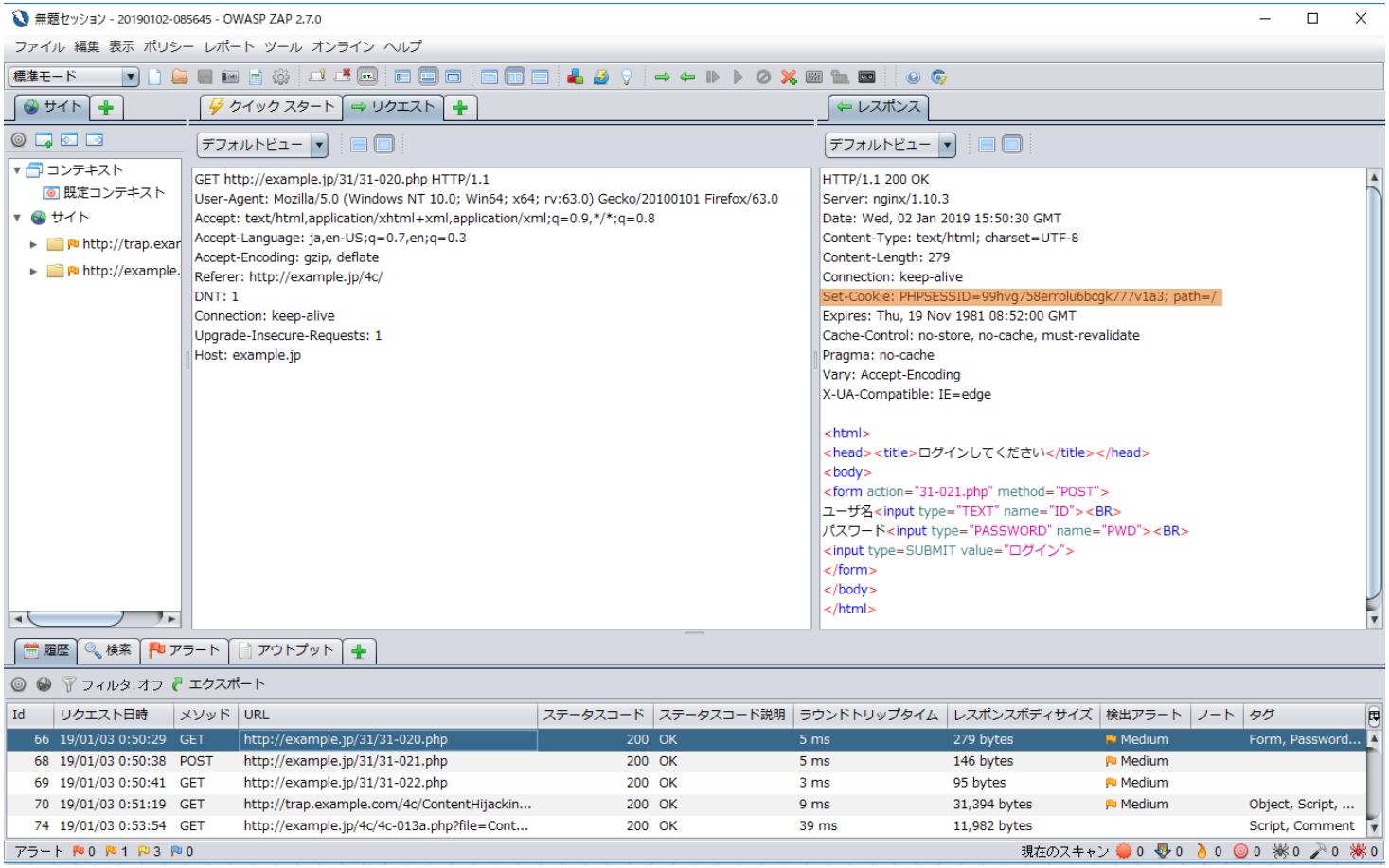
【サーバ：ContentHijackingLoader.html 】

仮想システム参照

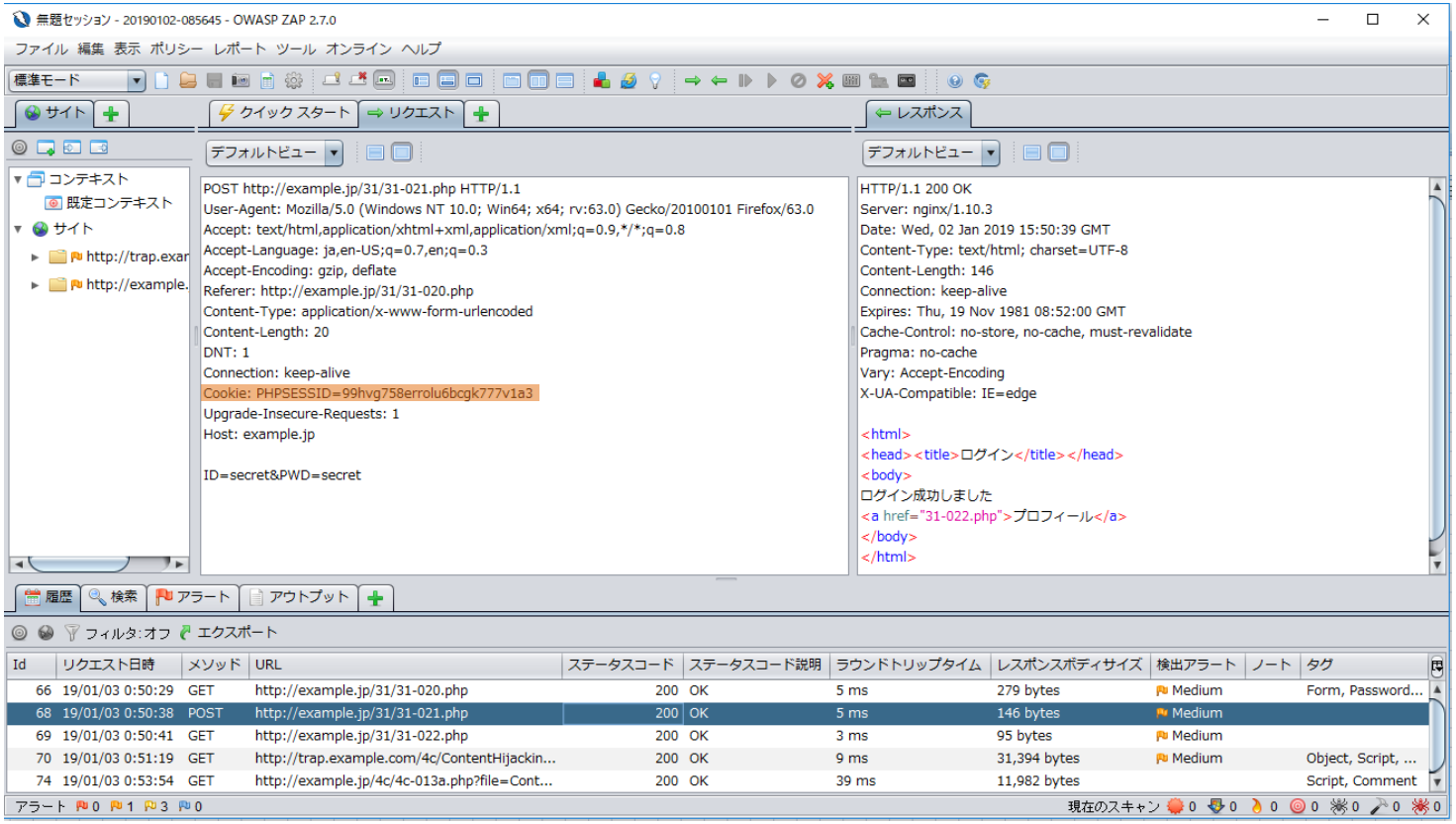
【サーバ：ContentHijackingLoader.pdf 】

仮想システム参照

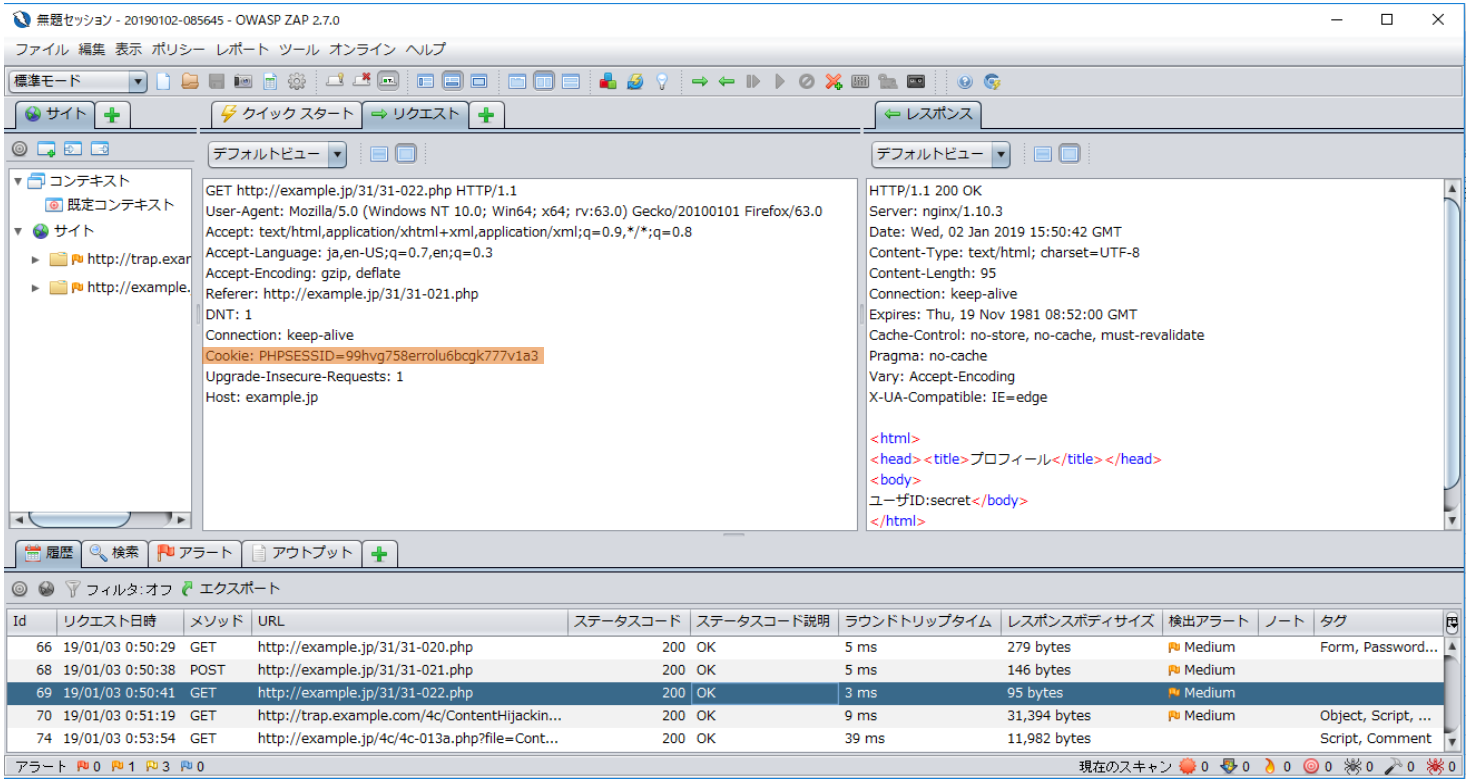
【ブラウザ→サーバ： リクエスト 31/31-020.php → レスポンス 】（firefoxでスニファー）



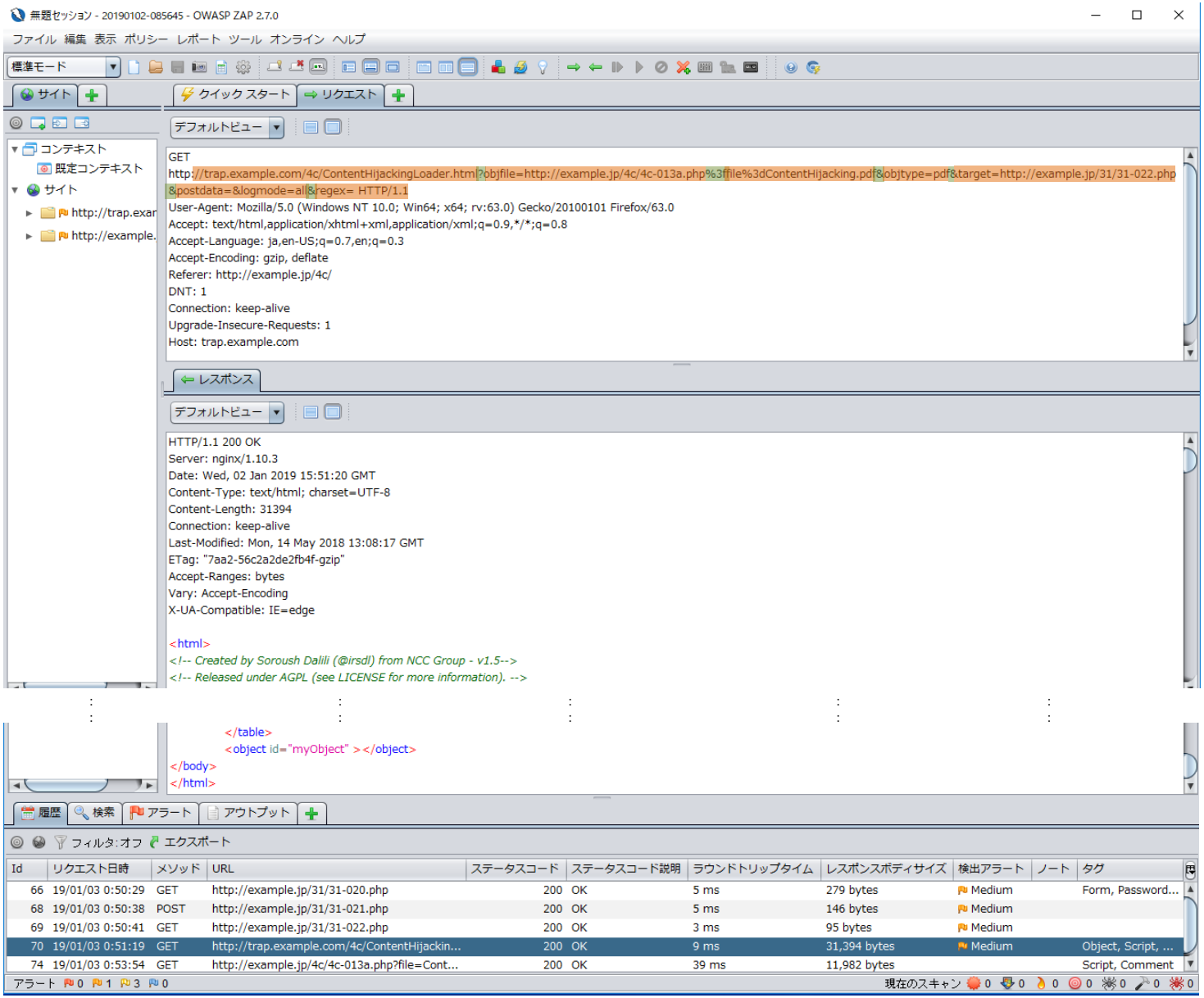
【ブラウザ→サーバ： リクエスト 31/31-021.php → レスポンス 】



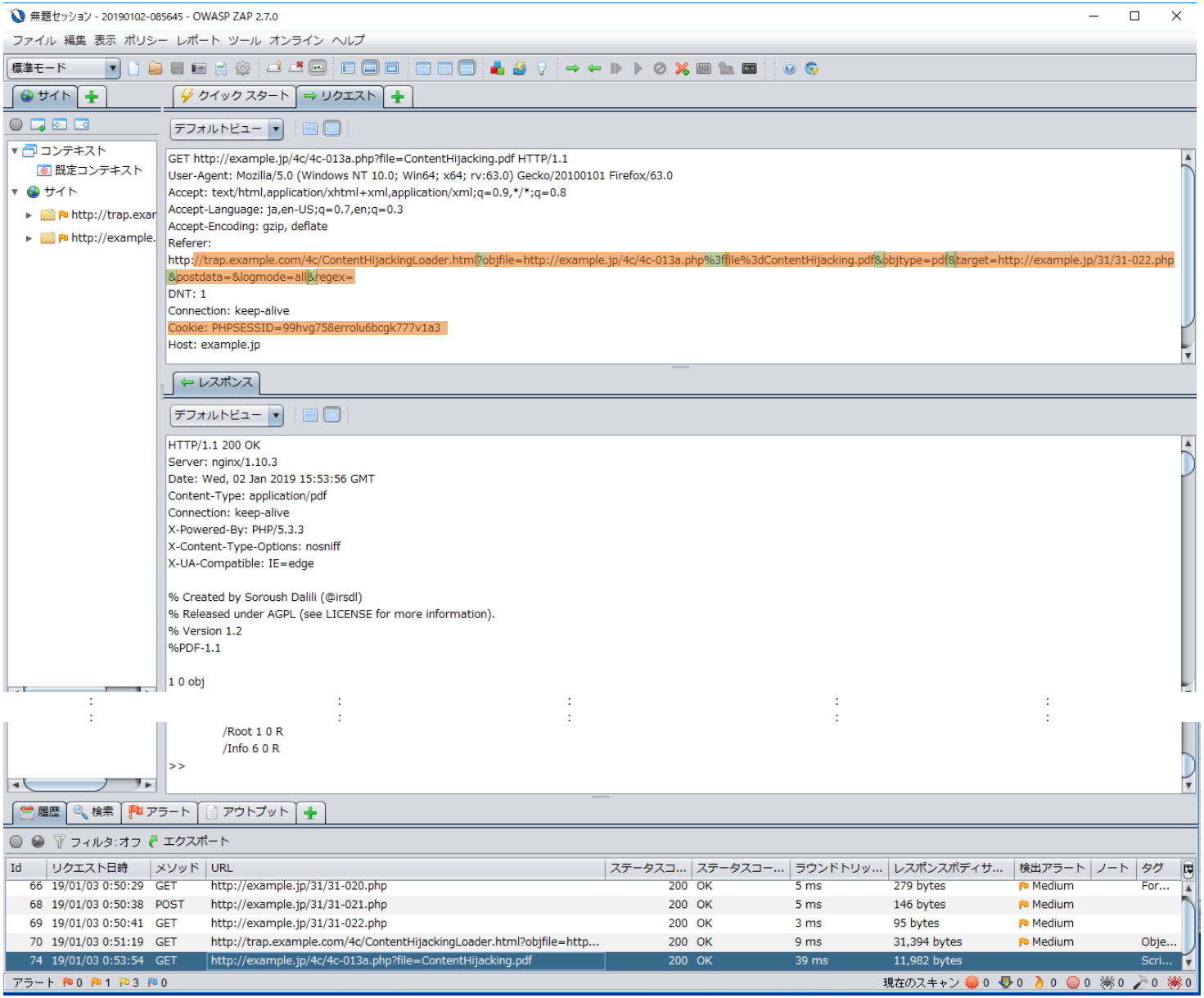
【ブラウザ→サーバ： リクエスト 31/31-022.php → レスポンス 】



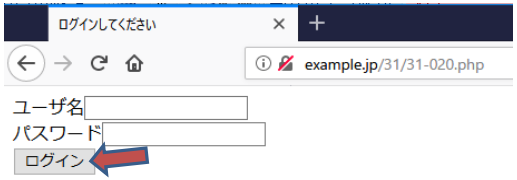
【ブラウザ→サーバ： リクエスト ContentHijackingLoader.html → レスポンス 】

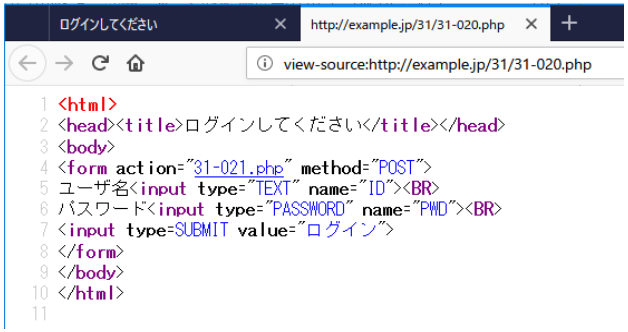


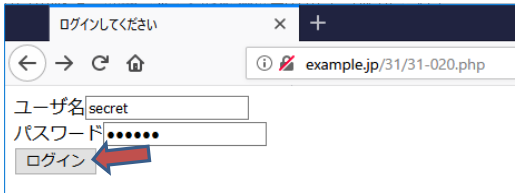
【ブラウザ→サーバ： リクエスト 4c/4c-013a.php → レスポンス 】

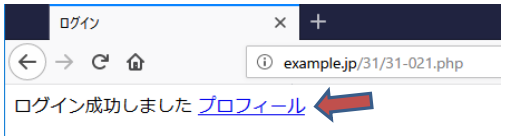


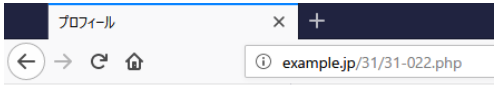
【ブラウザ】

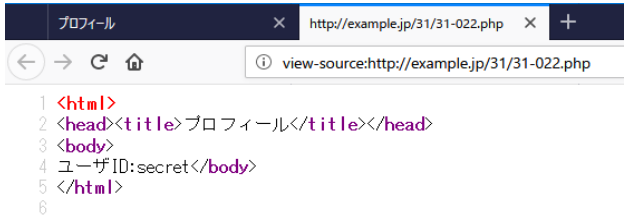








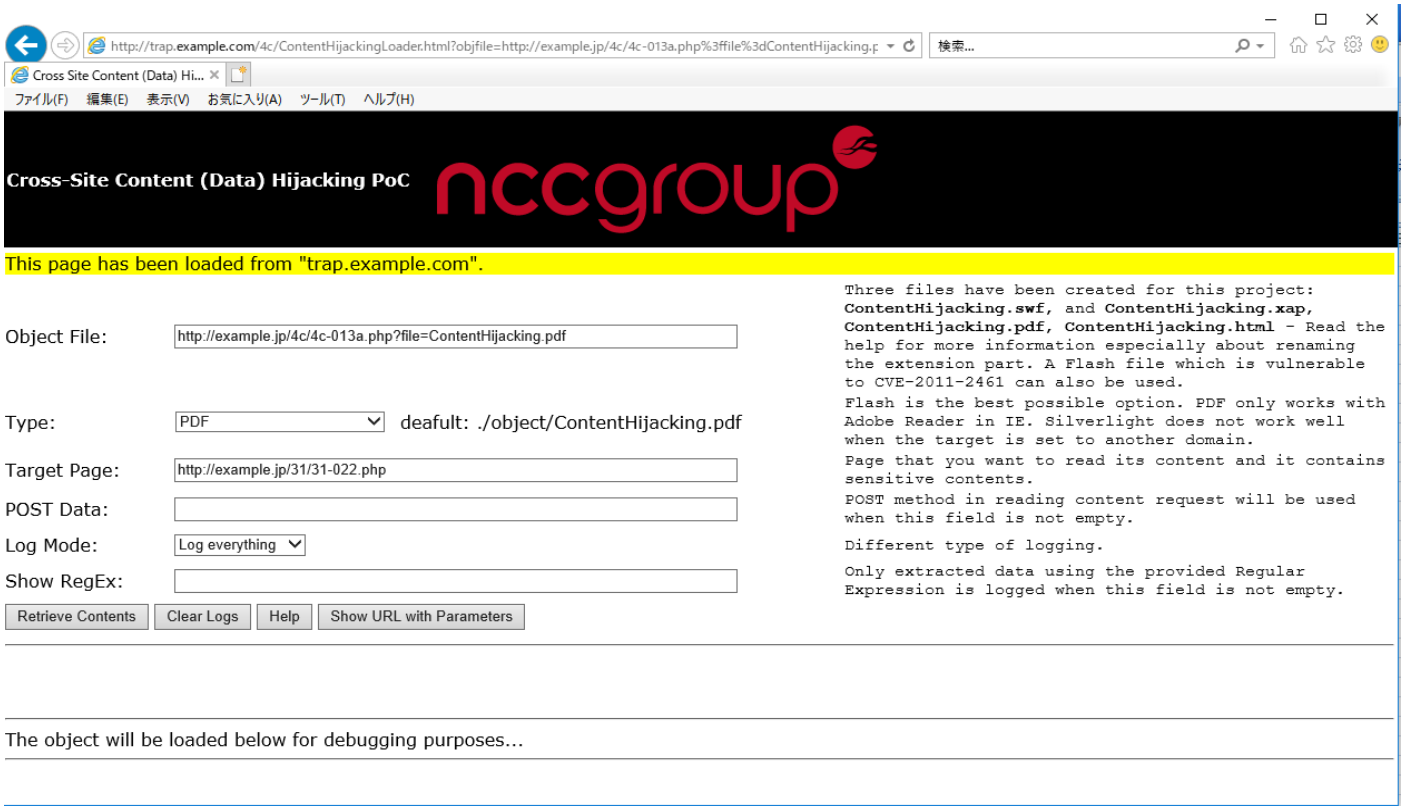


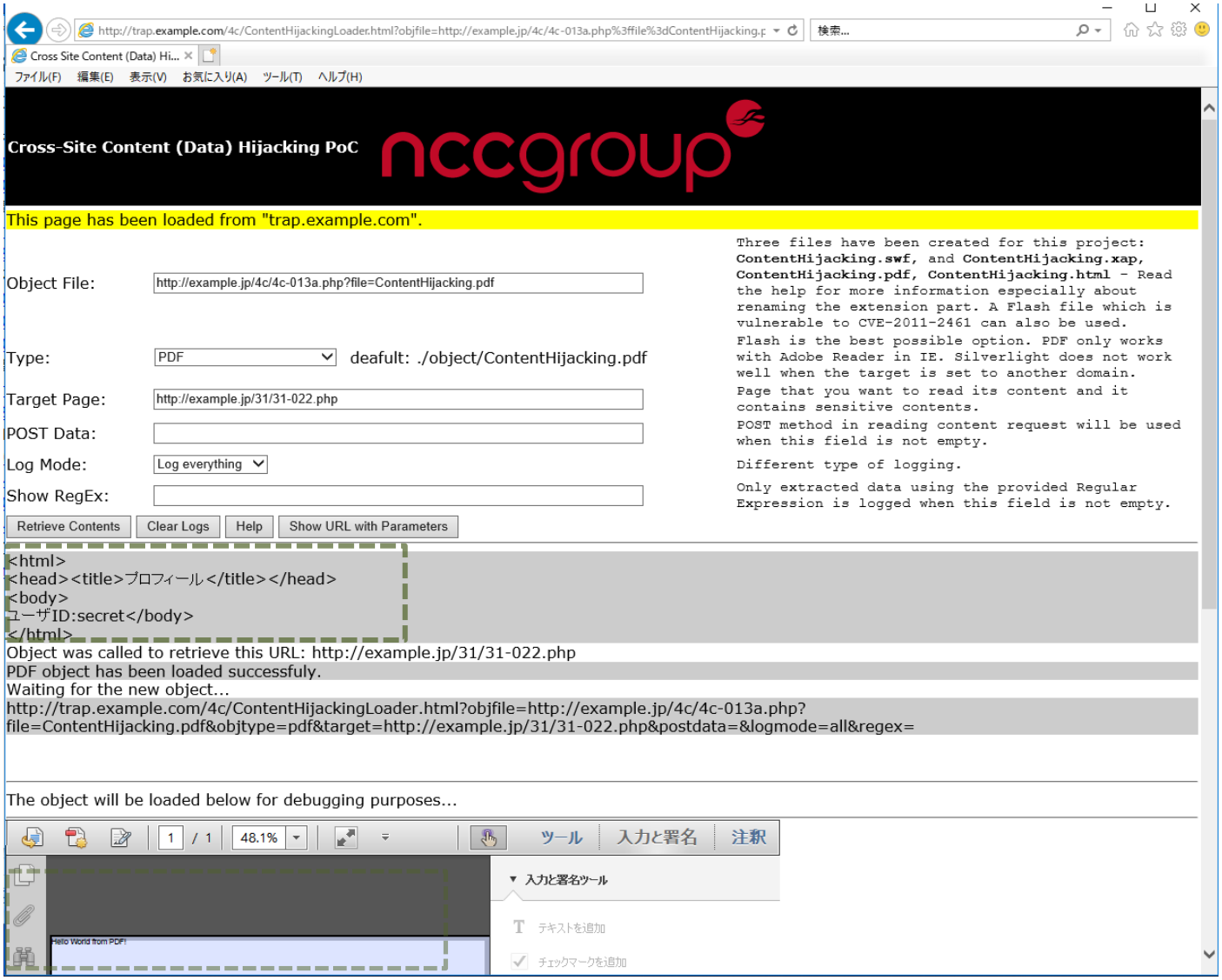


- 4.12.4 PDFのFormCalcによるコンテンツハイジャック(IEでアクセス)
  1. 31-020 :ログイン
  2. コンテンツハイジャック(PDF)
  3. 4c-011d:ファイルアップロード（FormCalc対策版）
  4. コンテンツハイジャック(PDF; 対策版)

  以下はコンテンツハイジャック攻撃用データ。
  1. ContentHijacking.pdf

InternetExploreで実行


http://trap.example.com/4c/ContentHijackingLoader.html?objfile=http://example.jp/4c/4c-013a.php%3ffile%3dContentHijacking.p
検索...
Cross Site Content (Data) Hi...
ファイル(F) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)
Cross-Site Content (Data) Hijacking PoC
nccgroup
This page has been loaded from "trap.example.com".
Object File:
http://example.jp/4c/4c-013a.php?file=ContentHijacking.pdf
Type:
PDF
deafult: ./object/ContentHijacking.pdf
Target Page:
http://example.jp/31/31-022.php
POST Data:
Log Mode:
Log everything
Show RegEx:
Retrieve Contents
Clear Logs
Help
Show URL with Parameters
Three files have been created for this project: ContentHijacking.swf, and ContentHijacking.xap, ContentHijacking.pdf, ContentHijacking.html - Read the help for more information especially about renaming the extension part. A Flash file which is vulnerable to CVE-2011-2461 can also be used.
Flash is the best possible option. PDF only works with Adobe Reader in IE. Silverlight does not work well when the target is set to another domain.
Page that you want to read its content and it contains sensitive contents.
POST method in reading content request will be used when this field is not empty.
Different type of logging.
Only extracted data using the provided Regular Expression is logged when this field is not empty.
The object will be loaded below for debugging purposes...



レスポンス の 31/31-022.php が参照できてしまう

4c-011d:ファイルアップロード（FormCalc対策版）

【ブラウザ】

- 4.12.4 PDFのFormCalcによるコンテンツハイジャック(IEでアクセス)
  1. 31-020 :ログイン
  2. コンテンツハイジャック(PDF)
  3. 4c-011d:ファイルアップロード（FormCalc対策版）
  4. コンテンツハイジャック(PDF; 対策版)

  以下はコンテンツハイジャック攻撃用データ。
  1. ContentHijacking.pdf

phpinfo
ホームに戻る

```
<li>4.12.4 PDFのFormCalcによるコンテンツハイジャック(IEでアクセス)</li>
<ol>
<li><a href="/31/31-020.php">31-020 :ログイン</a></li>
<li><a href="http://trap.example.com/4c/ContentHijackingLoader.html?objfile=http://example.jp/4c/4c-013a.php%3ffile%3dContentHijacking.pdf&objtype=pdf&target=http://example.jp/31/31-022.php&postdata=&logmode=all&regex=">コンテンツハイジャック(PDF)</a></li>
<li><a href="4c-011d.php">4c-011d:ファイルアップロード（FormCalc対策版）</a></li>
<li><a href="http://trap.example.com/4c/ContentHijackingLoader.html?objfile=http://api.example.net/4c/4c-013d.php%3ffile%3dContentHijacking.pdf&objtype=pdf&target=http://example.jp/31/31-022.php&postdata=&logmode=all&regex=">コンテンツハイジャック(PDF; 対策版)</a></li>
</ol>
以下はコンテンツハイジャック攻撃用データ。
<ol>
<li><a href="ContentHijacking.pdf" download="ContentHijacking.pdf">ContentHijacking.pdf</a></li>
</ol>
</ul>
<a href="phpinfo.php">phpinfo</a><br>
<a href="/">ホームに戻る</a>
</body>
</html>
```

【サーバ：4c/4c-011d.php 】

/var/www/html/4c/4c-011d.php - wasbook@example.jp - エディタ - WinSCP

```
<body>
<form action="4c-012d.php" method="post" enctype="multipart/form-data">
ファイル：  <input type="file" name="imgfile" size="20"><br>
<input type="submit" value="アップロード">
</form>
</body>
```

【サーバ：4c/4c-013d.php 】

/var/www/html/4c/4c-013d.php - wasbook@example.jp - エディタ - WinSCP

```
<?php
define('UPLOADPATH', '/var/upload');
$mimes = array('pdf' => 'application/pdf');
if ($_SERVER['REQUEST_METHOD'] !== 'POST') {
  header("HTTP/1.1 400 Bad Request");
  die('POSTメソッドでリクエストしてください');
}
$file = @$_POST['file'];
$orgfile = @$_POST['orgfile'];
$info = pathinfo($file);       // ファイル情報の取得
$ext = strtolower($info['extension']);     // 拡張子
$content_type = $mimes[$ext]; // Content-Typeの取得
if (! $content_type) {
  header("HTTP/1.1 400 Bad Request");
  die('拡張子はpdfを指定ください');
}
if (preg_match('/[\r\n]/', $orgfile) === 1) {
  header("HTTP/1.1 400 Bad Request");
  die('orgfileに改行を含めることはできません');
}
header('Content-Type: ' . $content_type);
header('X-Content-Type-Options: nosniff');
header('X-Download-Options: noopen');
header('Content-Disposition: attachment; filename="' . $orgfile  . '"');
readfile(UPLOADPATH . '/' . basename($file));
?>
```

【サーバ：4c/4c-012d.php 】

/var/www/html/4c/4c-012d.php - wasbook@example.jp - エディタ - WinSCP

```
<?php
define('UPLOADPATH', '/var/upload');
function get_upload_file_name($tofile) {
  // 拡張子のチェック
  $info = pathinfo($tofile);
  $ext = strtolower($info['extension']);
  if ($ext != 'pdf') {
    die('拡張子はpdfを指定ください');
  }
  // 以下、ユニークなファイル名の生成
  $count = 0; // ファイル名作成試行の回数
  do {
    // ファイル名の組み立て
    $file = sprintf('%s/%08x.%s', UPLOADPATH, mt_rand(), $ext);
    // ファイルを作成する。既存の場合はエラー
    $fp = @fopen($file, 'x');
  } while ($fp === FALSE && ++$count < 10);
  if ($fp === FALSE) {
    die('ファイルが作成できません');
  }
  fclose($fp);
  return $file;
}
$tmpfile = $_FILES["imgfile"]["tmp_name"];
$orgfile = $_FILES["imgfile"]["name"];
if (! is_uploaded_file($tmpfile)) {
  die('ファイルがアップロードされていません');
}
$tofile = get_upload_file_name($orgfile);
if (! move_uploaded_file($tmpfile, $tofile)) {
  die('ファイルをアップロードできません');
}
// $imgurl = '4c-013d.php?file=' . basename($tofile);
?>
<body>
<?php echo htmlspecialchars($orgfile, ENT_NOQUOTES, 'UTF-8'); ?>をアップロードしました</a><BR>
<form action='4c-013d.php' method='post'>
<input type="hidden" name="file" value="<?php echo htmlspecialchars(basename($tofile)); ?>">
<input type="hidden" name="orgfile" value="<?php echo htmlspecialchars(basename($orgfile)); ?>">
<input type="submit" value="ダウンロード">
</form>
</body>
```

【サーバ：ContentHijackingLoader.html 】　　**【サーバ：ContentHijackingLoader.pdf 】**

仮想システム参照　　仮想システム参照

レスポンスヘッダ X-Contnt-Type-Options：nosniff の設定

IEには、Content-Typeの解釈が曖昧なところがあり、その改善でマイクロソフトはこのレスポンスヘッダを実装した。その後、IE以外のブラウザにも実装された。
指定すると、Content-Typeからのみコンテンツのファイル種類を判断します。そのため、text/htmlでなければJavaScriptの実行までに至りません。

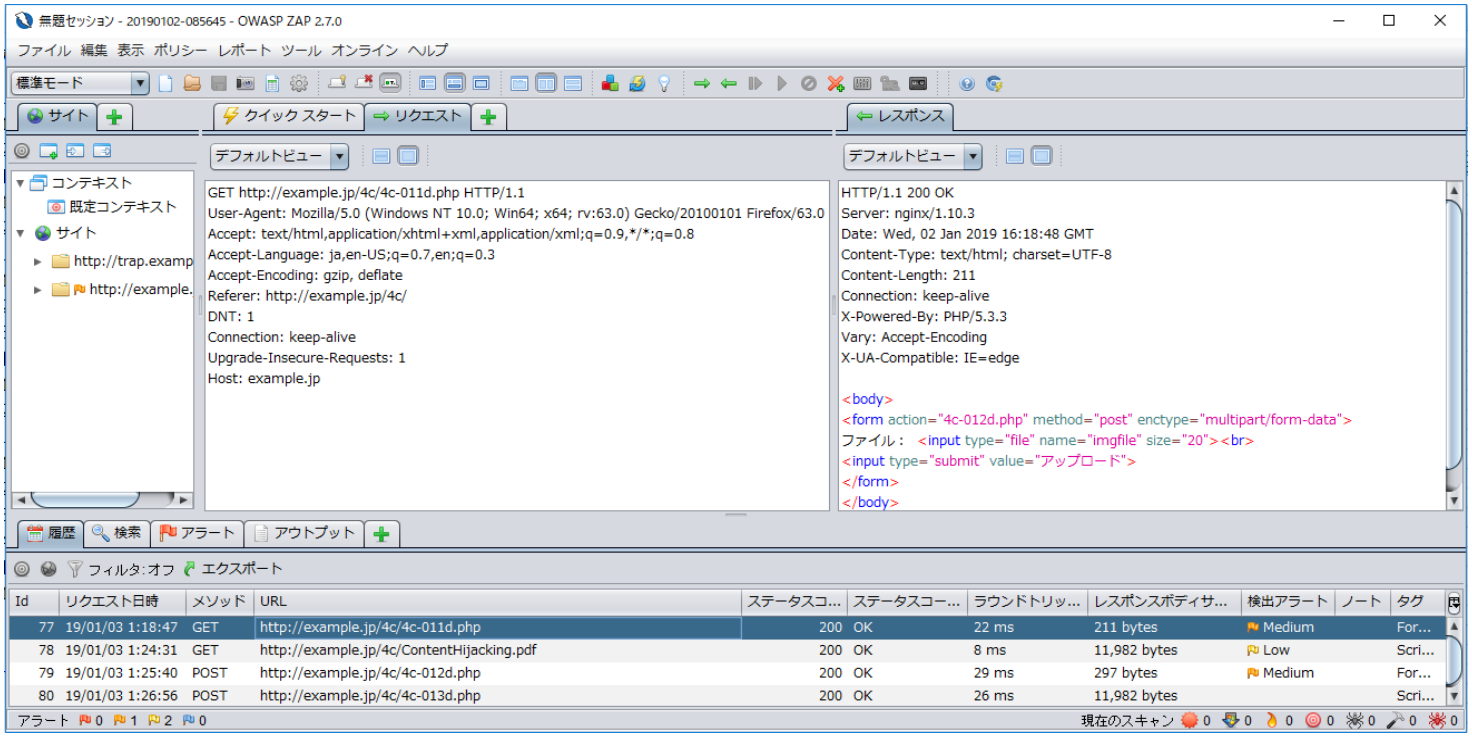
レスポンスヘッダ Content-Disposition の設定

inline（デフォルト値。ウェブページの一部として、またはウェブページとして表示可能であることを示します）、もしくは attachment（ダウンロードすべきであることを示します。
多くのブラウザは filename パラメーターの値を使い、「名前を付けて保存」ダイアログを表示します）を最初のパラメーターとして指定します。

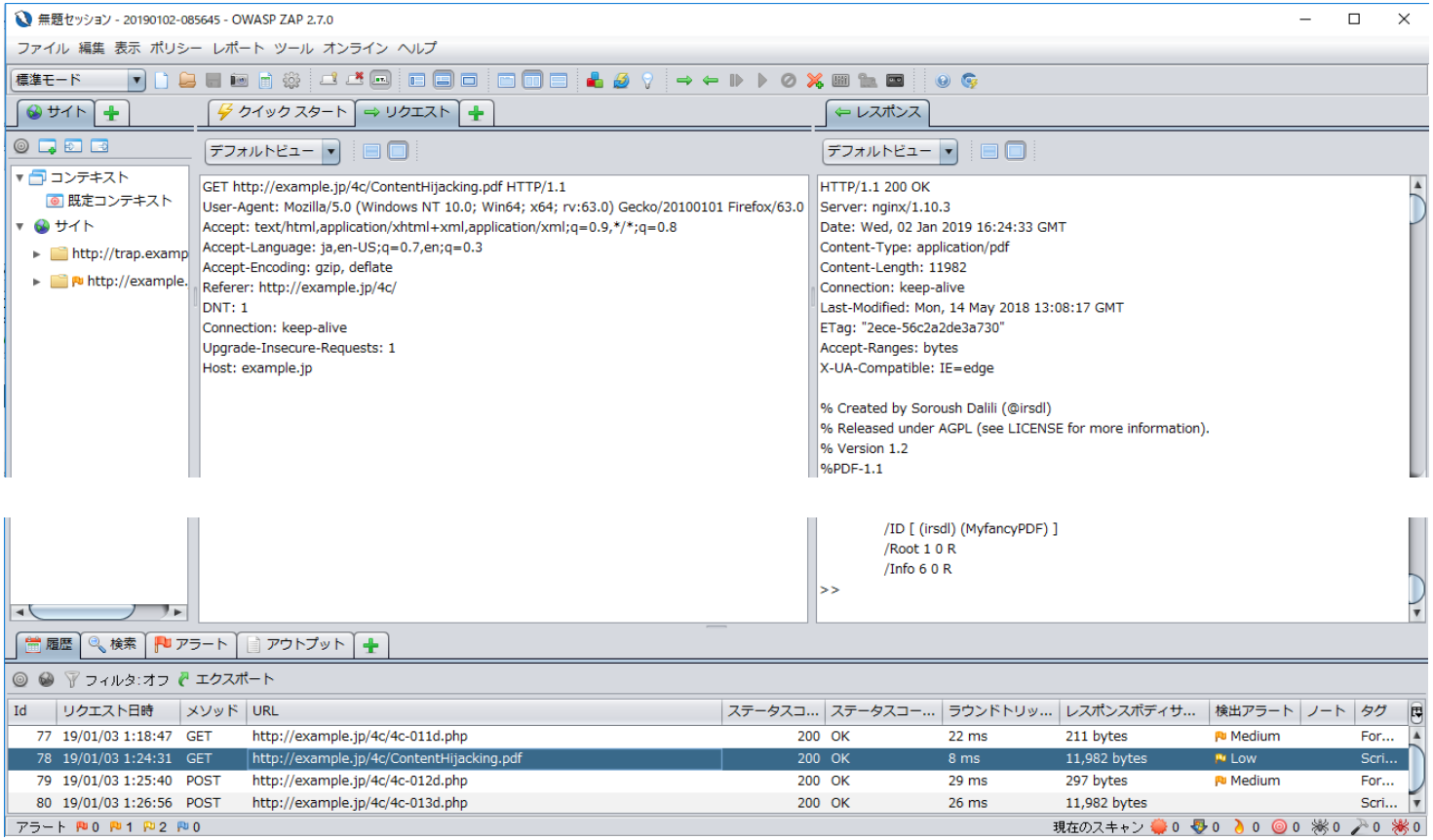
レスポンスヘッダ X-Download-Options の設定

IEでダウンロードしたファイルを直接開かせない。保存をした後でなければファイルの内容が参照できなくなります。（IE固有の仕様）

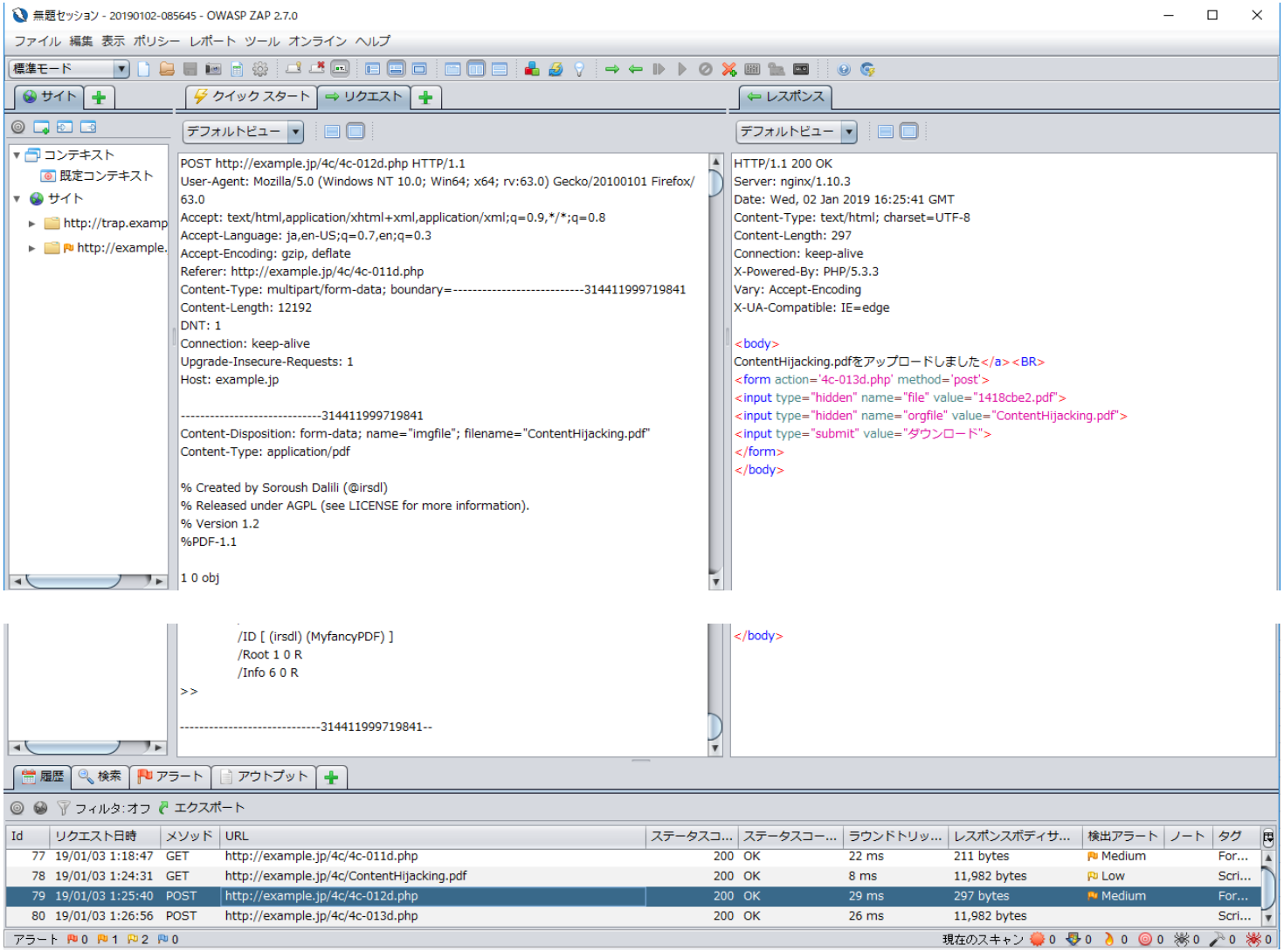
【ブラウザ→サーバ： リクエスト 4c/4c-011d.php → レスポンス 】（firefoxでスニファー）



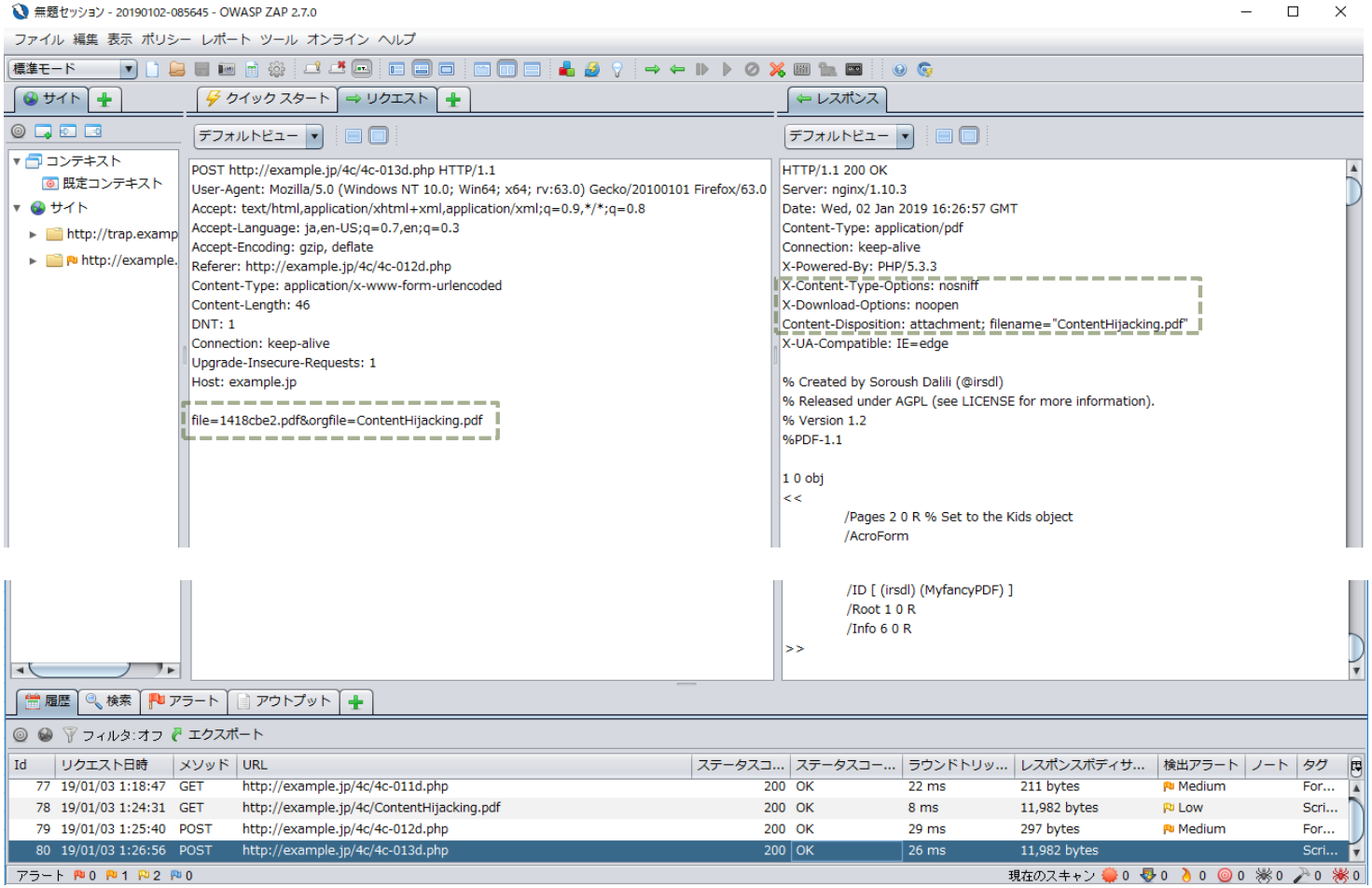
【ブラウザ→サーバ： リクエスト ContentHijackingLoader.pdf → レスポンス 】



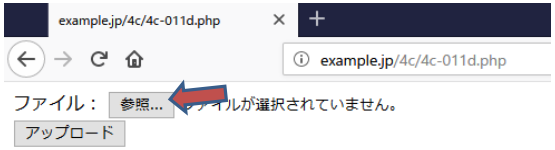
【ブラウザ→サーバ： リクエスト 4c/4c-012d.php → レスポンス 】

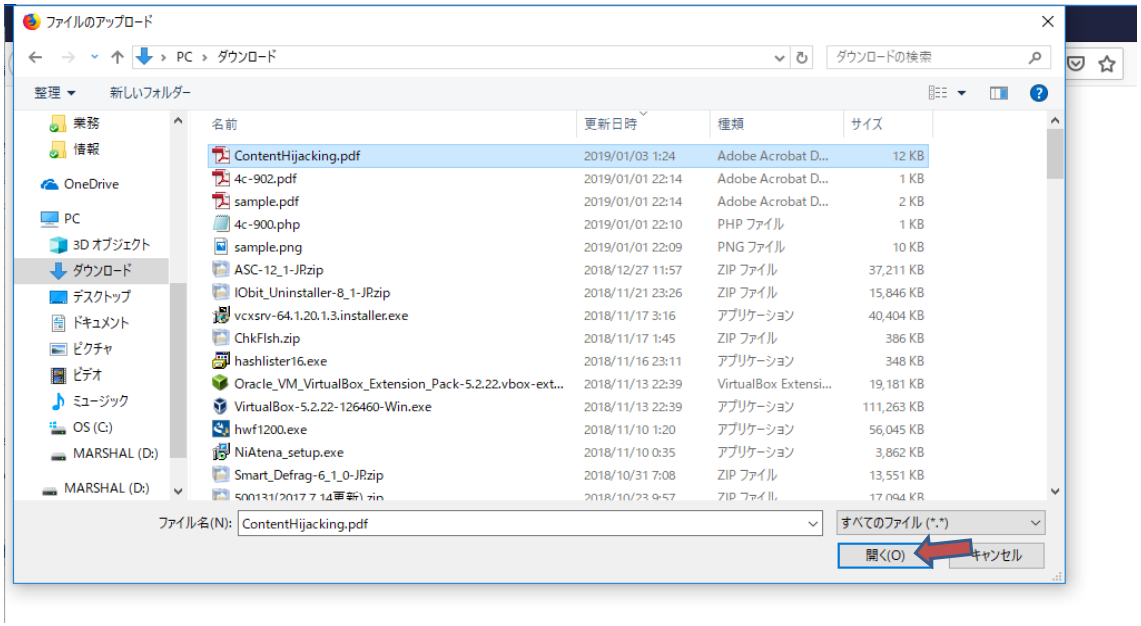


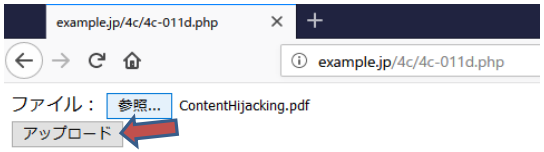
【ブラウザ→サーバ： リクエスト 4c/4c-013d.php → レスポンス 】

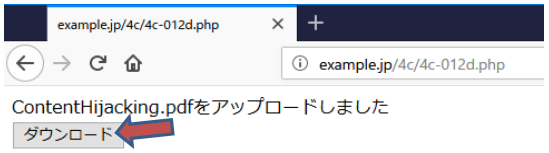


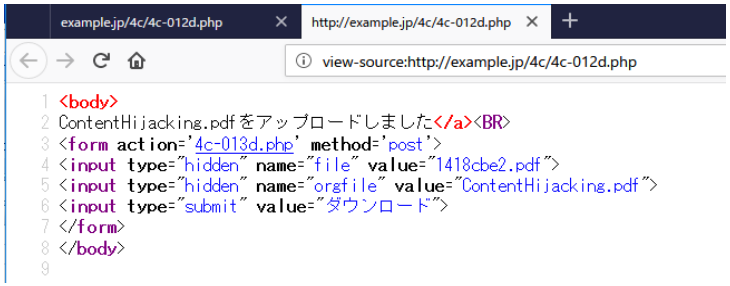
【ブラウザ】（firefoxの場合）

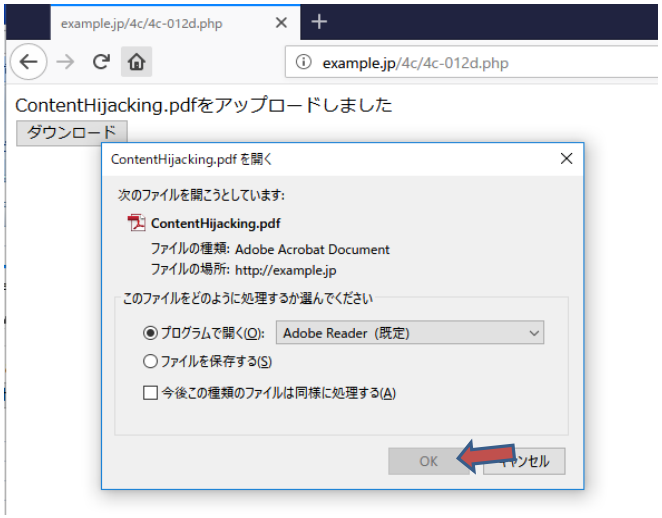


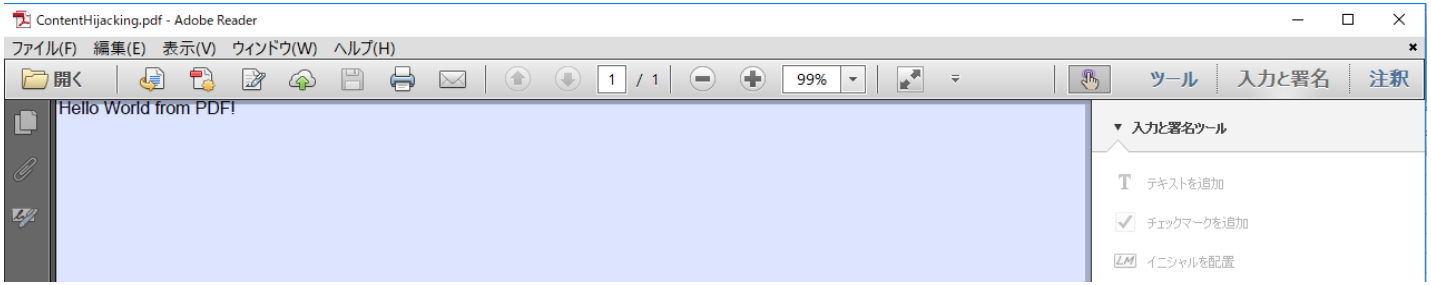


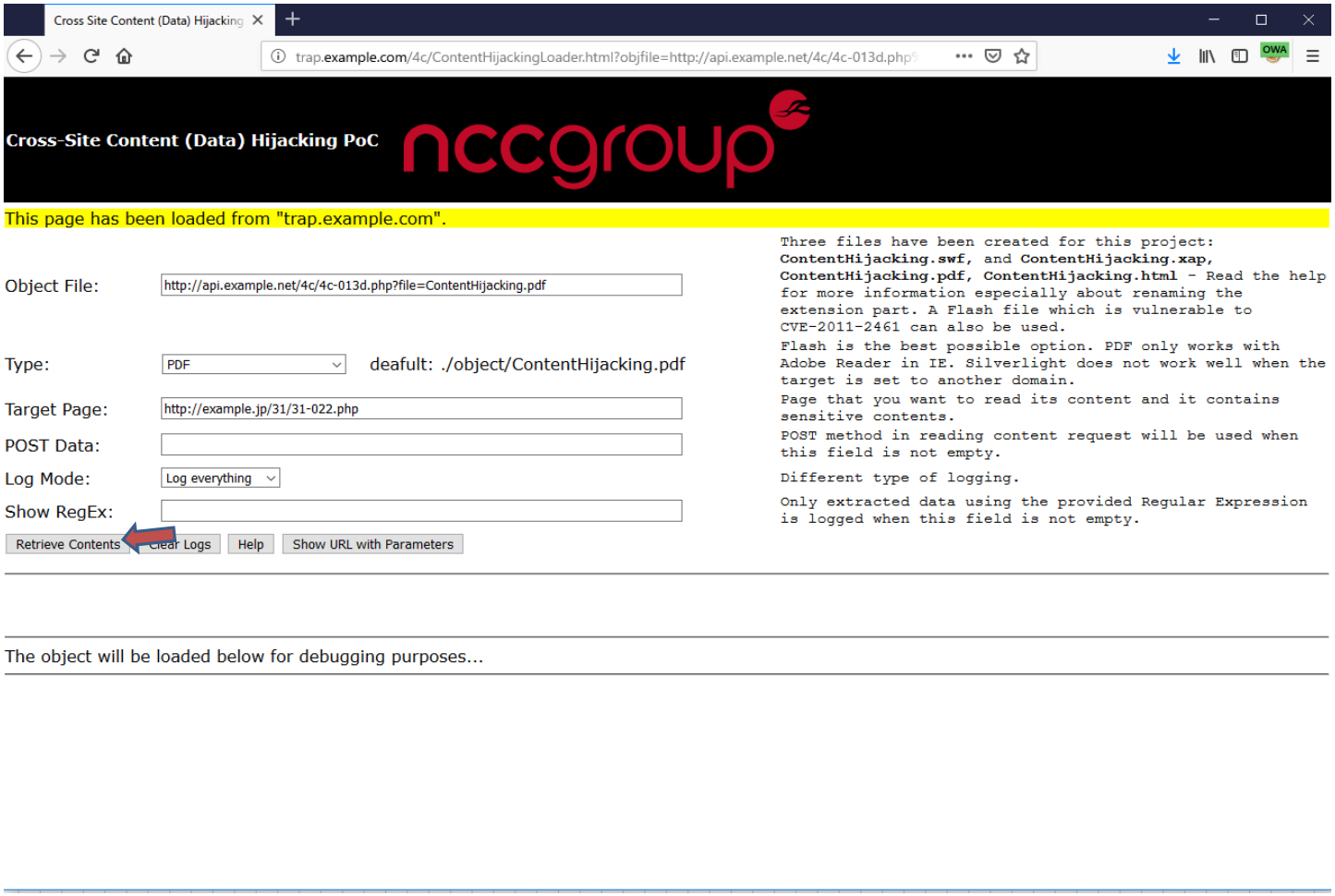


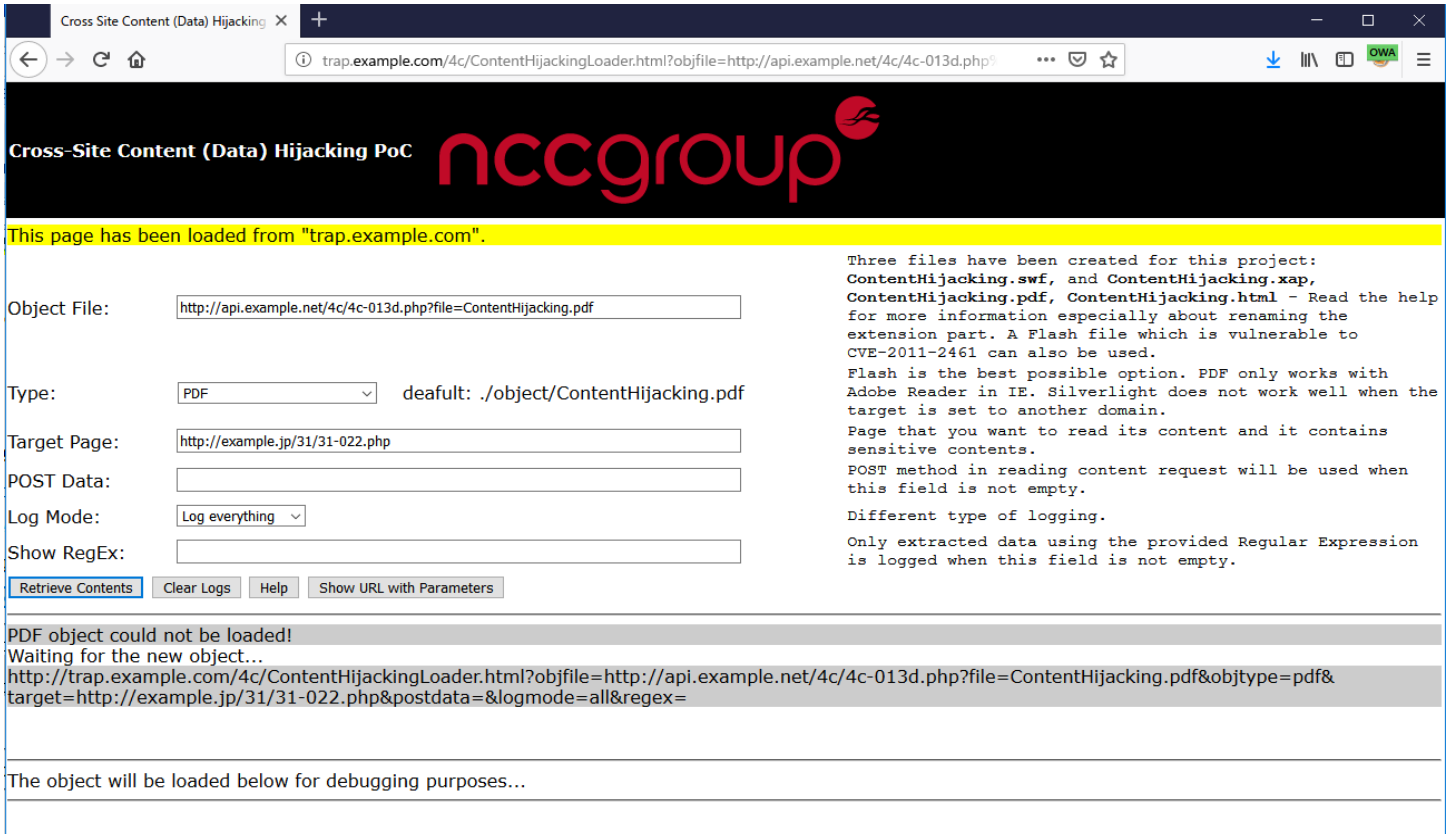





example.jp/4c/4c-012d.php
example.jp/4c/4c-012d.php
ContentHijacking.pdfをアップロードしました
ダウンロード
ContentHijacking.pdf を開く
次のファイルを開こうとしています:
ContentHijacking.pdf
ファイルの種類: Adobe Acrobat Document
ファイルの場所: http://example.jp
このファイルをどのように処理するか選んでください
プログラムで開く(O):
Adobe Reader (既定)
ファイルを保存する(S)
今後この種類のファイルは同様に処理する(A)
OK

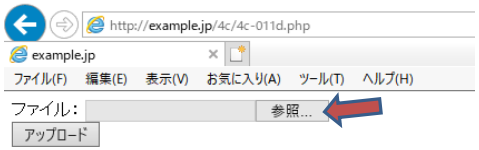

ContentHijacking.pdf - Adobe Reader
ファイル(F) 編集(E) 表示(V) ウィンドウ(W) ヘルプ(H)
開く
1 / 1
99%
ツール
入力と署名
注釈
Hello World from PDF!
入力と署名ツール
テキストを追加
チェックマークを追加
イニシャルを配置

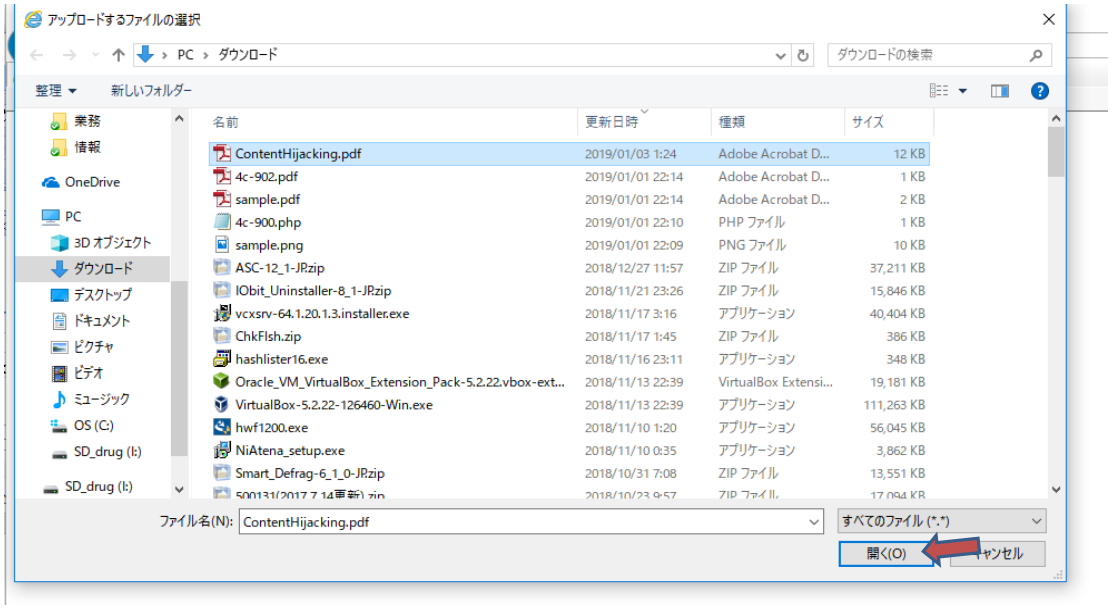

Cross Site Content (Data) Hijacking
trap.example.com/4c/ContentHijackingLoader.html?objfile=http://api.example.net/4c/4c-013d.php
Cross-Site Content (Data) Hijacking PoC
nccgroup
This page has been loaded from "trap.example.com".
Object File:
http://api.example.net/4c/4c-013d.php?file=ContentHijacking.pdf
Type:
PDF
deafult: ./object/ContentHijacking.pdf
Target Page:
http://example.jp/31/31-022.php
POST Data:
Log Mode:
Log everything
Show RegEx:
Retrieve Contents
Clear Logs
Help
Show URL with Parameters
Three files have been created for this project: ContentHijacking.swf, and ContentHijacking.xap, ContentHijacking.pdf, ContentHijacking.html - Read the help for more information especially about renaming the extension part. A Flash file which is vulnerable to CVE-2011-2461 can also be used.
Flash is the best possible option. PDF only works with Adobe Reader in IE. Silverlight does not work well when the target is set to another domain.
Page that you want to read its content and it contains sensitive contents.
POST method in reading content request will be used when this field is not empty.
Different type of logging.
Only extracted data using the provided Regular Expression is logged when this field is not empty.
The object will be loaded below for debugging purposes...

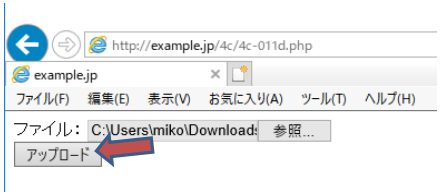


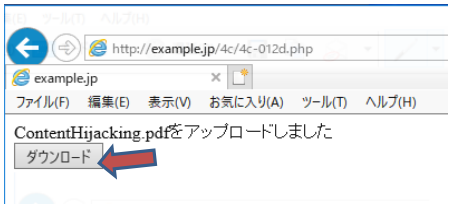
レスポンス の 31/31-022.php が参照できなくなりました

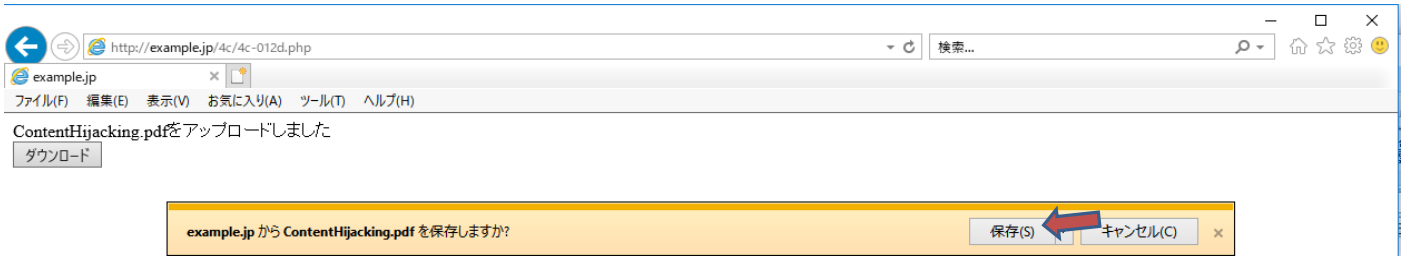
【ブラウザ】（InternetExplore11の場合）

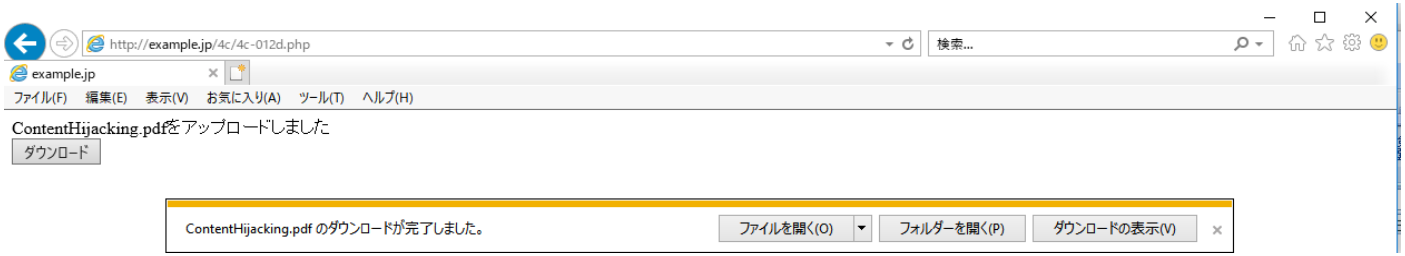


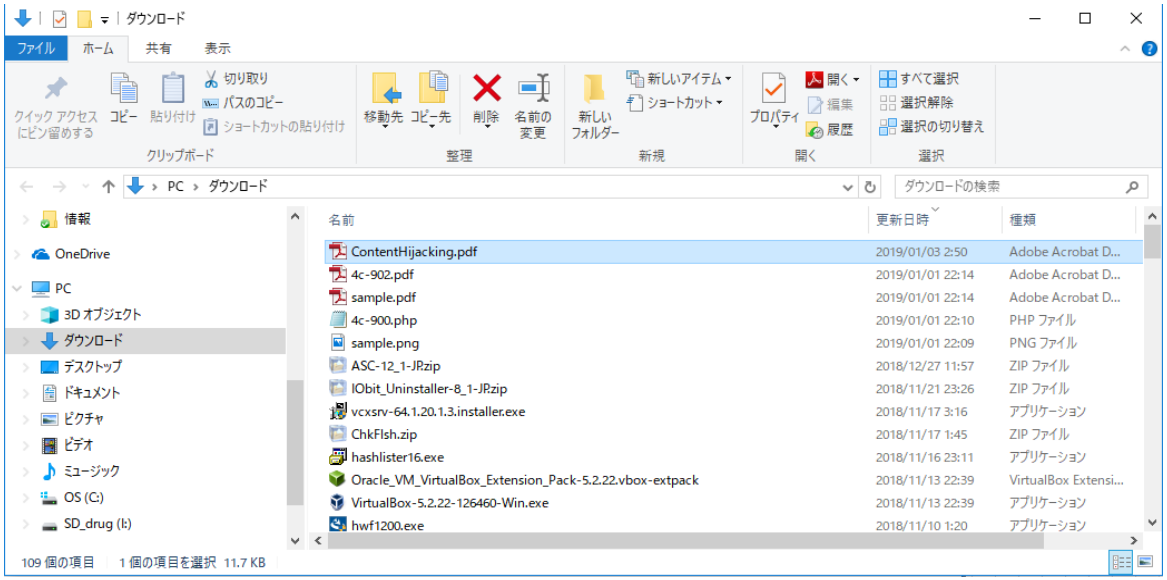







http://example.jp/4c/4c-012d.php
検索...
example.jp
ファイル(F) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)
ContentHijacking.pdfをアップロードしました
ダウンロード
example.jp から ContentHijacking.pdf を保存しますか?
保存(S)
キャンセル(C)


http://example.jp/4c/4c-012d.php
検索...
example.jp
ファイル(F) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)
ContentHijacking.pdfをアップロードしました
ダウンロード
ContentHijacking.pdf のダウンロードが完了しました。
ファイルを開く(O)
フォルダーを開く(P)
ダウンロードの表示(V)

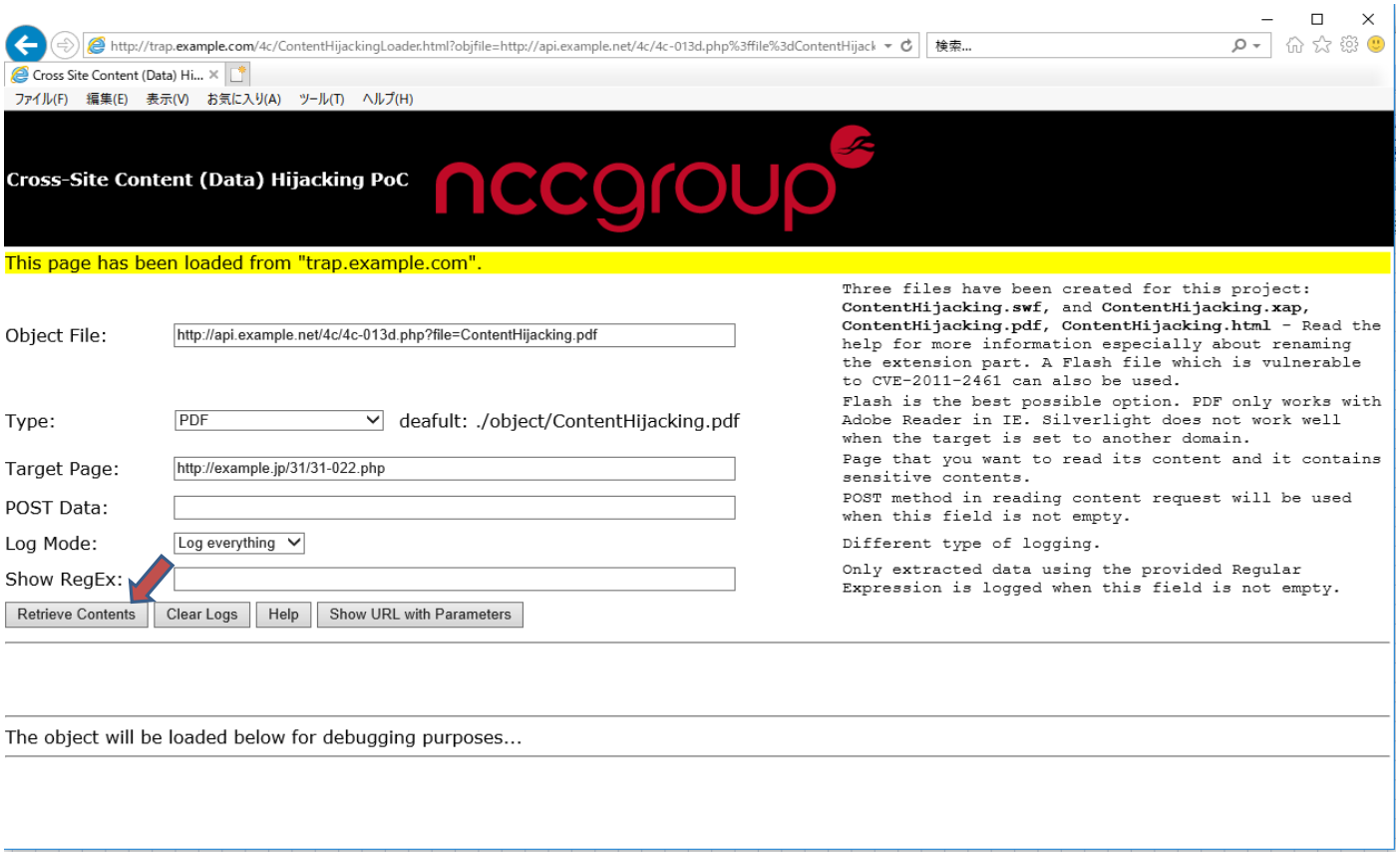

ダウンロード
ファイル ホーム 共有 表示
クイック アクセスにピン留めする
コピー
貼り付け
切り取り
パスのコピー
ショートカットの貼り付け
クリップボード
移動先
コピー先
削除
名前の変更
整理
新しいフォルダー
新しいアイテム
ショートカット
新規
プロパティ
開く
編集
履歴
開く
すべて選択
選択解除
選択の切り替え
選択
PC › ダウンロード
ダウンロードの検索
情報
OneDrive
PC
3D オブジェクト
ダウンロード
デスクトップ
ドキュメント
ピクチャ
ビデオ
ミュージック
OS (C:)
SD_drug (I:)
名前
更新日時
種類
ContentHijacking.pdf 2019/01/03 2:50 Adobe Acrobat D...
4c-902.pdf 2019/01/01 22:14 Adobe Acrobat D...
sample.pdf 2019/01/01 22:14 Adobe Acrobat D...
4c-900.php 2019/01/01 22:10 PHP ファイル
sample.png 2019/01/01 22:09 PNG ファイル
ASC-12_1-JP.zip 2018/12/27 11:57 ZIP ファイル
IObit_Uninstaller-8_1-JP.zip 2018/11/21 23:26 ZIP ファイル
vcxsrv-64.1.20.1.3.installer.exe 2018/11/17 3:16 アプリケーション
ChkFlsh.zip 2018/11/17 1:45 ZIP ファイル
hashlister16.exe 2018/11/16 23:11 アプリケーション
Oracle_VM_VirtualBox_Extension_Pack-5.2.22.vbox-extpack 2018/11/13 22:39 VirtualBox Extensi...
VirtualBox-5.2.22-126460-Win.exe 2018/11/13 22:39 アプリケーション
hwf1200.exe 2018/11/10 1:20 アプリケーション
109 個の項目
1 個の項目を選択 11.7 KB

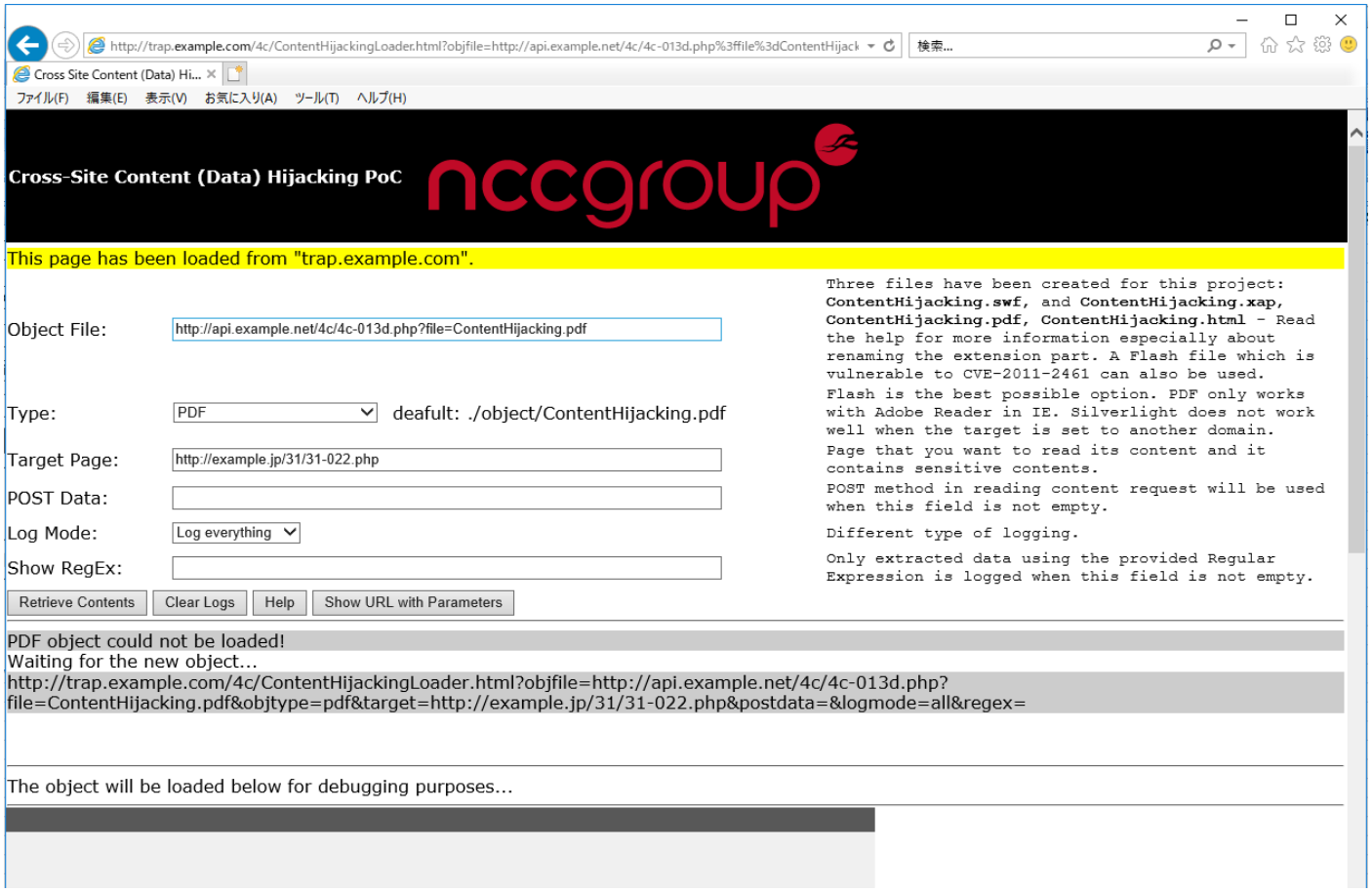
- 4.12.4 PDFのFormCalcによるコンテンツハイジャック(IEでアクセス)
  1. 31-020 :ログイン
  2. コンテンツハイジャック(PDF)
  3. 4c-011d:ファイルアップロード（FormCalc対策版）
  4. コンテンツハイジャック(PDF; 対策版)

InternetExploreで実行

以下はコンテンツハイジャック攻撃用データ。

1. ContentHijacking.pdf





レスポンス の 31/31-022.php が参照できなくなりました