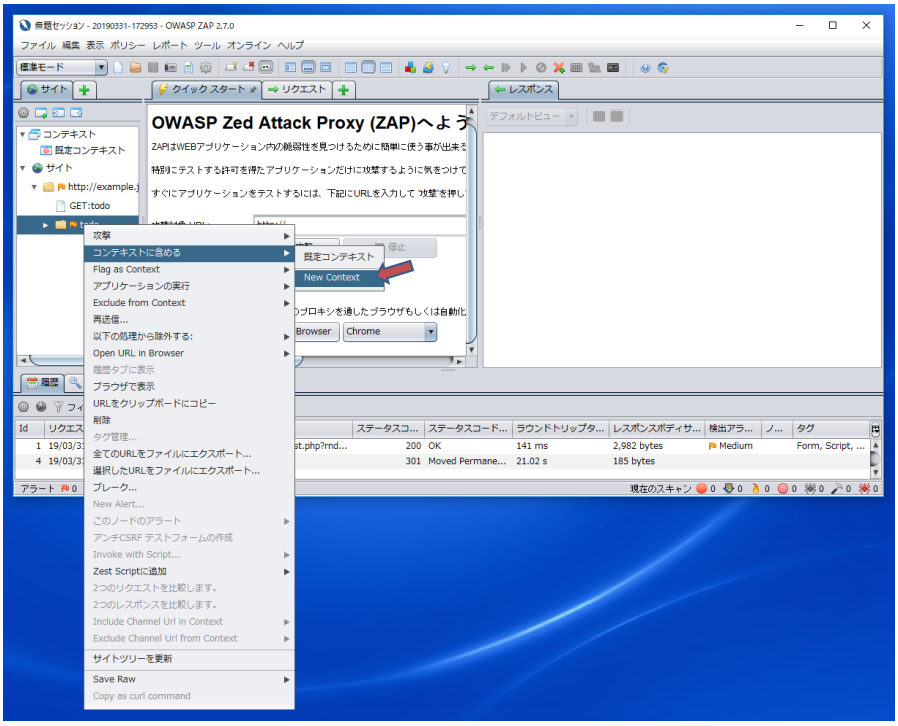
7.6 OWASP ZAP による自動脆弱性スキャン

OWASP ZAPの設定

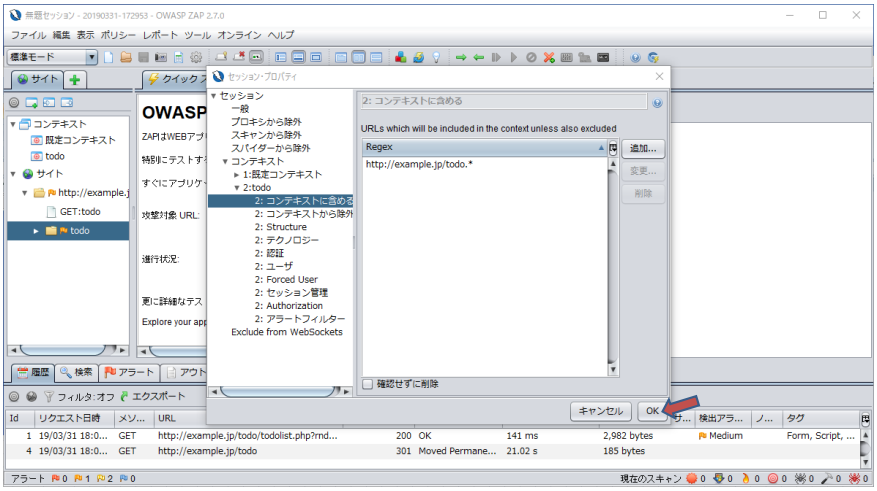
Firefoxにより診断対象アプリケーション（http://example.jp/todo/）にアクセスします。

OWASP ZAPに、診断対象アプリケーションの設定を行う

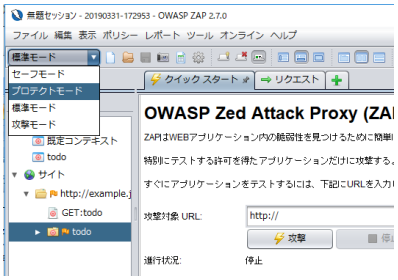


診断対象ディレクトリ「Todo」
→「コンテキストに含める」→「New Context」
をクリック

「2：コンテキストに含める」を選択し、「http://example.jp/todo.*」の表示を確認し、「OK」をクリック



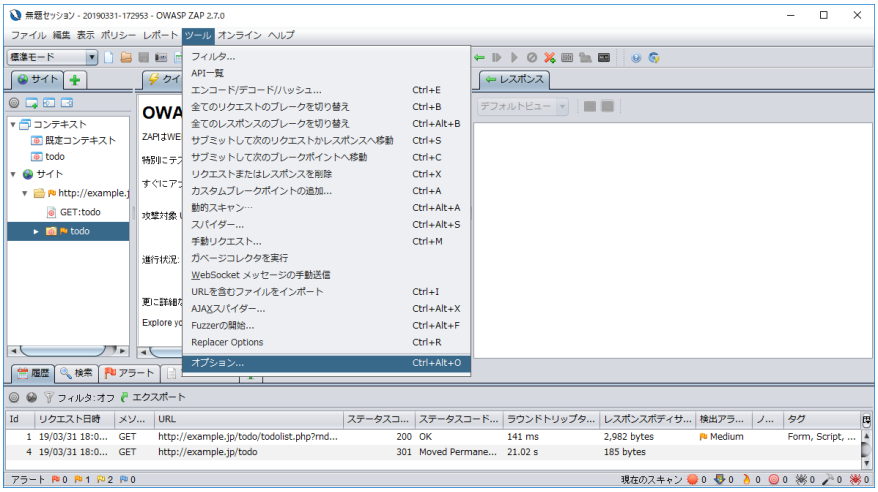
画面左上の「標準モード」を「プロテクトモード」に変更する



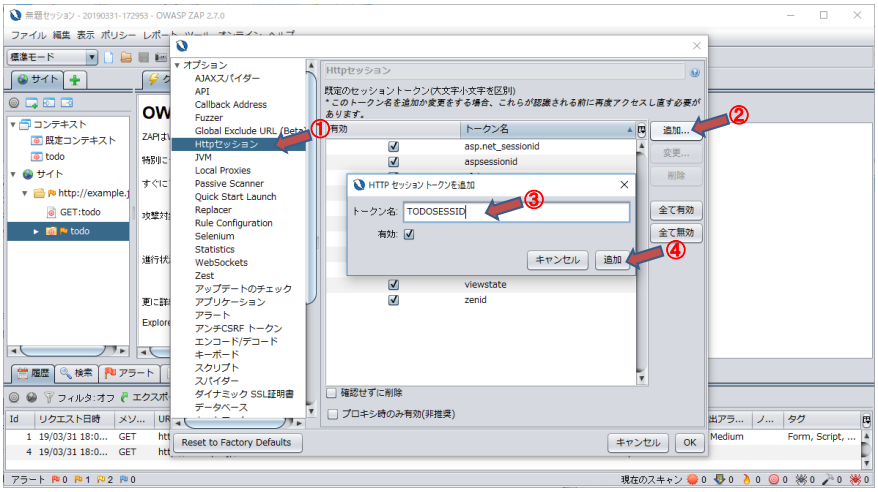
プロテクトモードを選択すると、ZAPは現在のコンテキストのみ診断します。
他サイトを意図せず攻撃してしまう事態を回避できます。

プロテクトモードではブレークポイント操作なども、コンテキスト指定されたURLのみ適応されます。

セッションID名の設定　　セッション追跡のため、セッションIDを含むクッキーの名前を登録します。



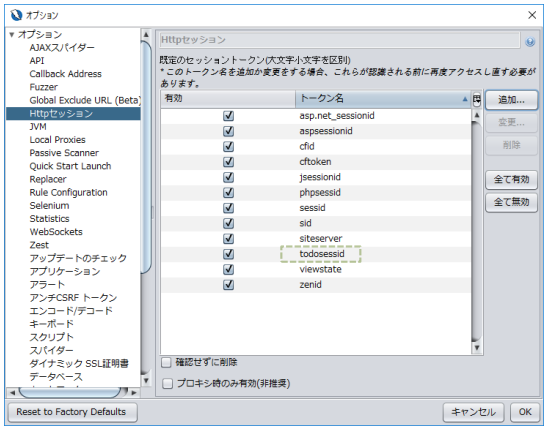
「ツール」→「オプション」をクリック



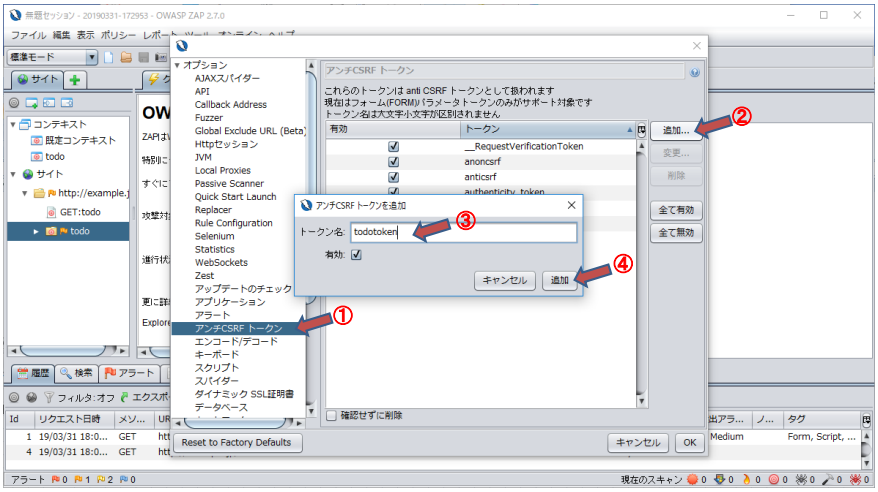
「Httpセッション」を選択し、

「追加」→トークン名に「TODOSESSID」を入力し、

「追加」をクリック



CSRF対策用トークン名の登録　「todotoken」がCSRF対策用トークンの名前のようなので、これを登録します。



同様に

「アンチCSRFトークン」を選択し、

「追加」→トークン名に「todotoken」を入力し、

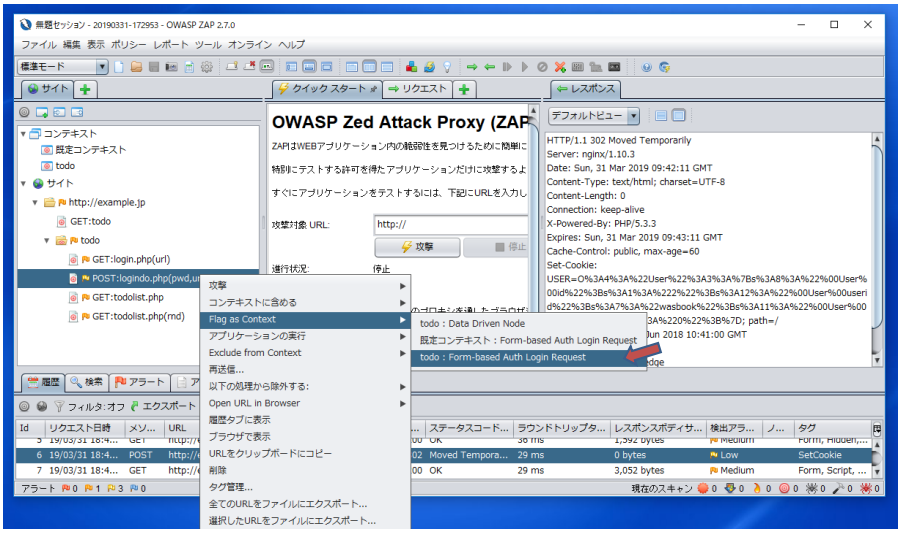
「追加」をクリック

セッション情報の設定

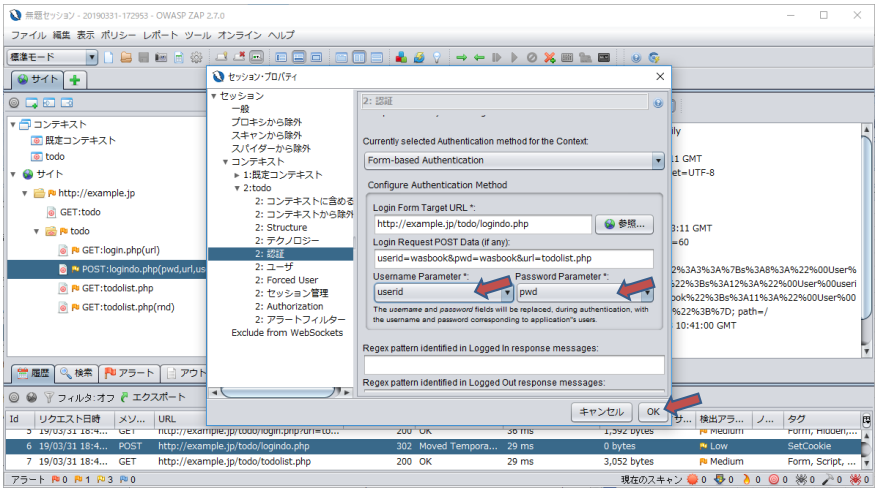
自動ログインの設定

Bad Todoの「ログイン」タブをクリックし、ID「wasbook」、PW「wasbook」でログインする

OWASP ZAP　のページ一覧、あるいは履歴からログインのPOSTリクエストを選択し、コンテキストメニューから以下操作します。



「POST : logindo.php(pwd,url,userid) → 「Flag as Context」→「todo : Form-based Auth Login Request」をクリック



Username Parameterに「userid」

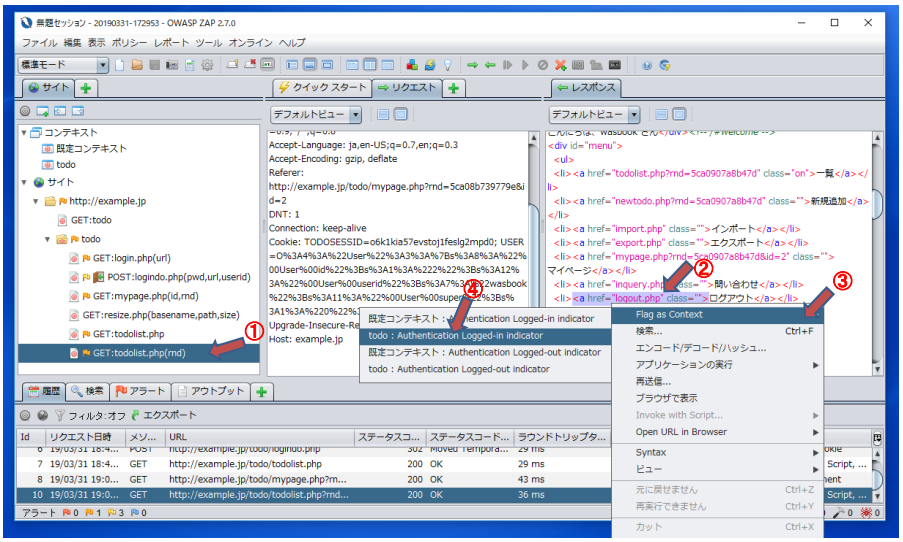
Password Parameterに「pwd」

を設定

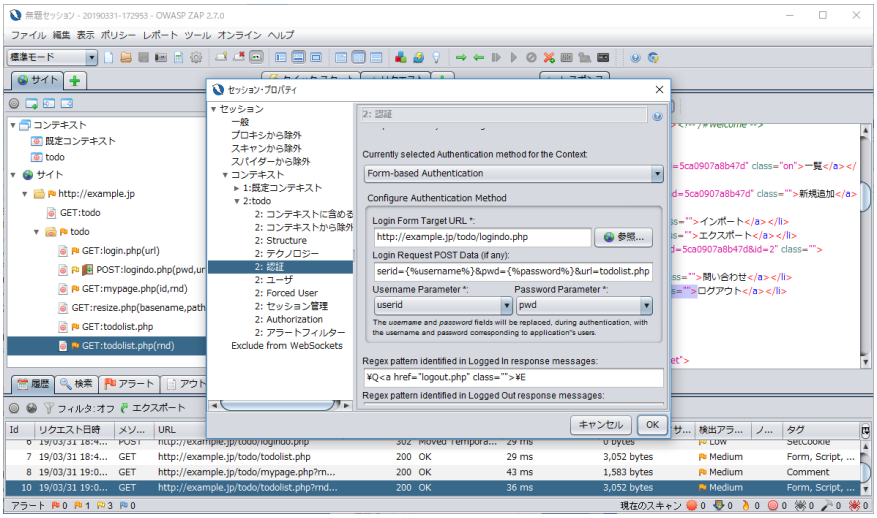
「OK」をクリック

ログイン状態を検出する設定　　OWASP ZAP　にログインに成功したことを検出できるように設定します。

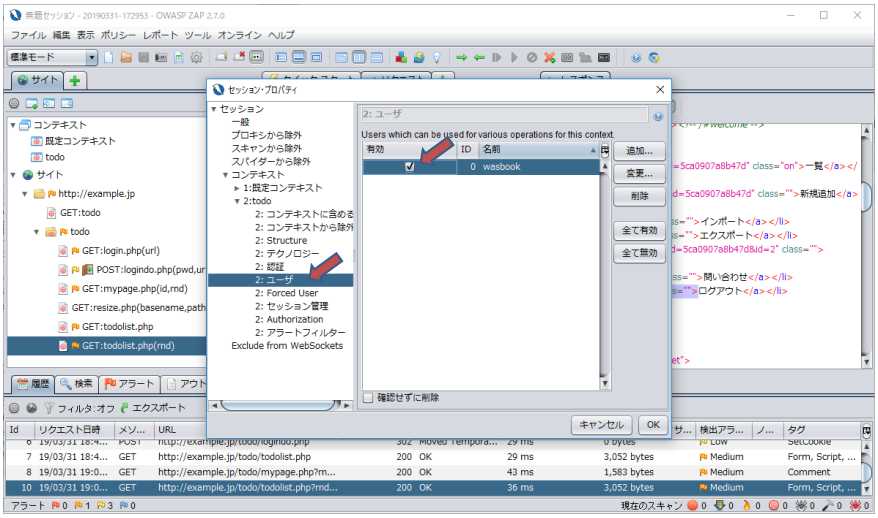
ログイン後のページtodolist.phpのリクエストを選択し、右側ペインの「レスポンス」タブから、「<a href ="logout.php>」を選択します。
さらに、「Flag as Context」-「todo : Authentication Logged-in indicator 」を選びます。



ログイン成功を検出する正規表現パターンが設定された

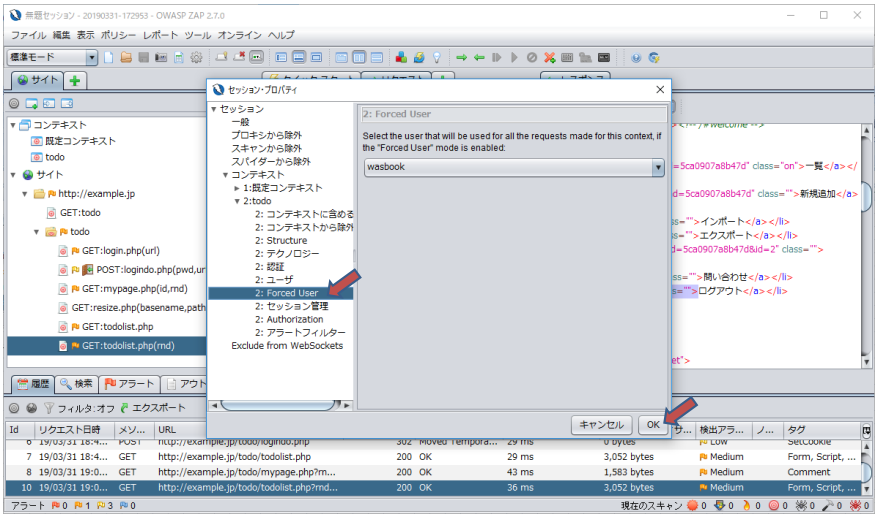


「OK」を押さずに、ユーザ選択に進みます。



「ユーザ」をクリックし、自動ログインユーザ名「wasbook」の有効をチェックします。

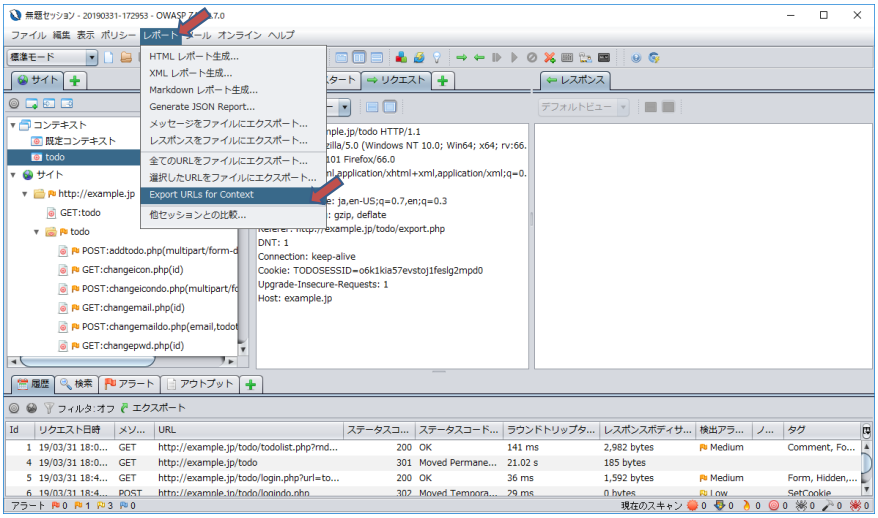
強制ログインユーザの選択　「Force User」をクリックし、「wasbook」が選択されているのを確認します。

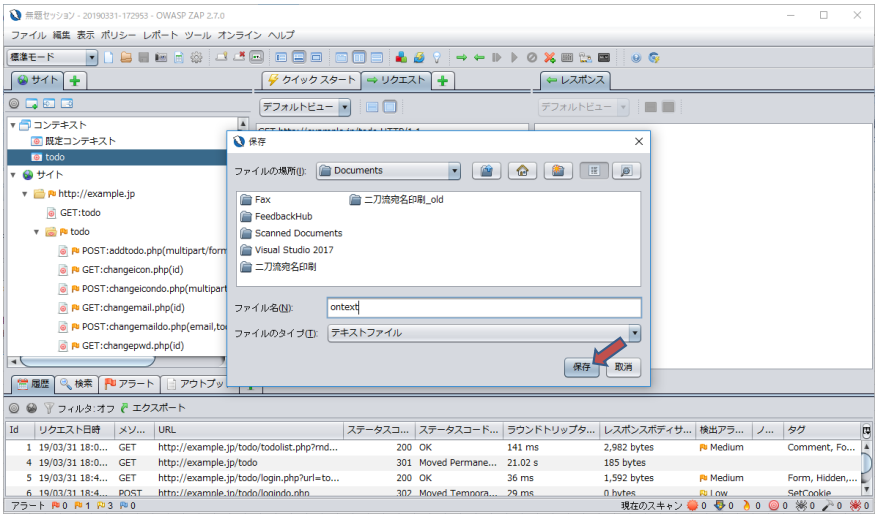


「OK」をクリックします。

クローリング

クローリングとは、Webページを巡回して、サイトのページ構成をZAPに記憶させることです。OWASP ZAP　には、スパイダーという自動クローリングの機能を持っていますが、手動実行した方が良い。
クローリング前の準備は設定しているので、クローリング作業の基本はすべてのページを閲覧することです。クローリングにはURLだけではなく、パラメータも記憶するので、良い診断のためには
すべてのパラメータが出現するように意識して巡回する必要があります。
たとえば、検索機能では、検索文字を指定するシーケンスと一覧リンクからの遷移ではパラメータが違ってきますので、両方の遷移が登録されるようにする必要があります。
クローリングが終了したら、コンテキストに含まれるURLの一覧をテキストファイルにエクスポートして、設計書などと比較し、クローリング結果の妥当性を確認することができます。

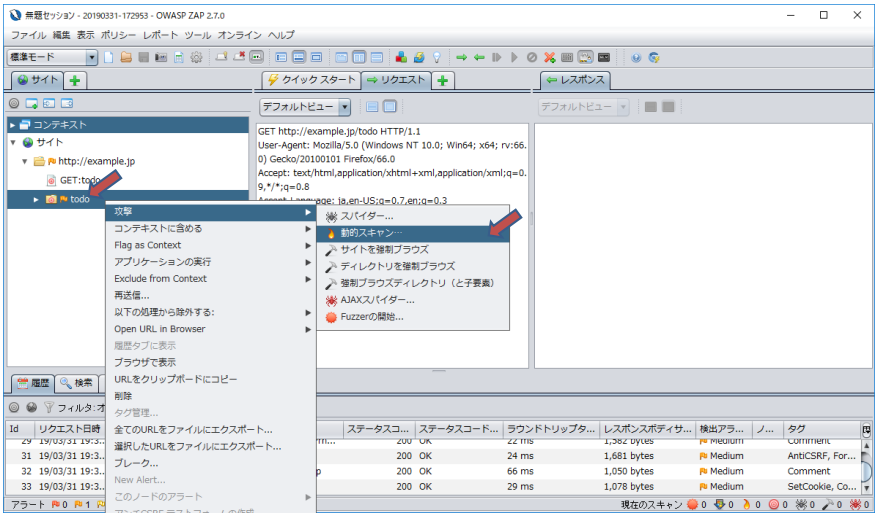




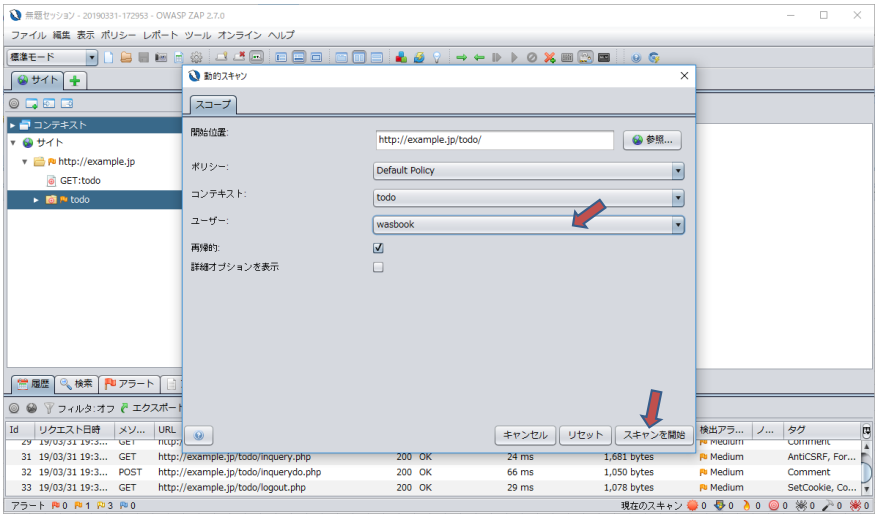
自動診断

錠前アイコンを閉じた状態にして、強制ログインモードを選択し、自動診断できます。

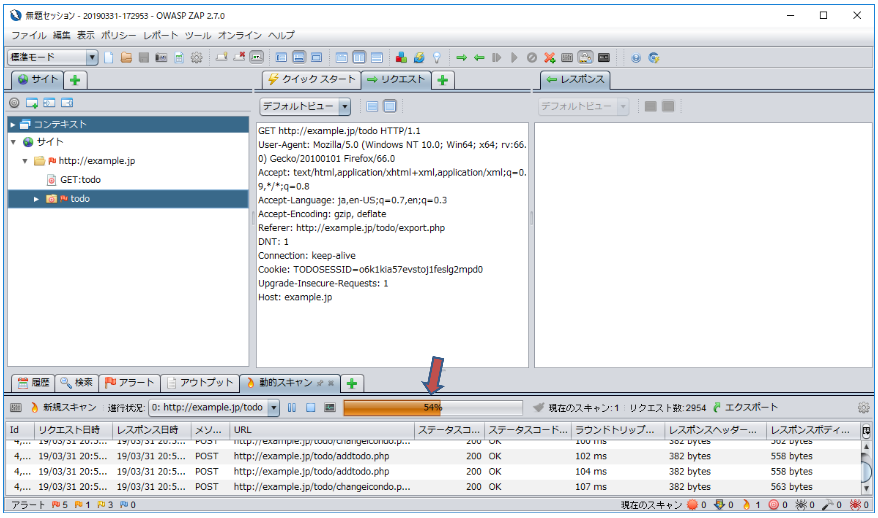
「todo」フォルダを選択し、コンテキストメニューから、「攻撃」→「動的スキャン」をクリック



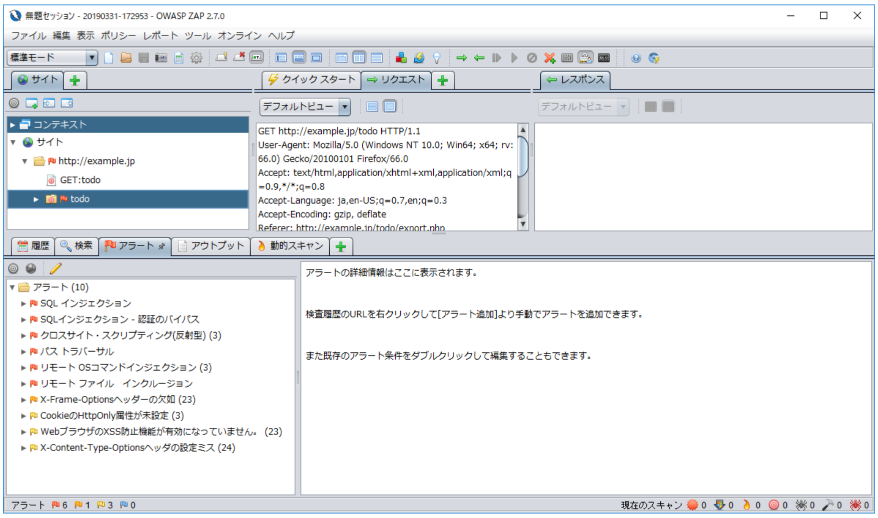
ユーザ「wasbook」を選択し、「スキャンを開始」をクリック



「詳細オプションを表示」をクリックすると動的スキャンをカスタマイズできるがここでは行わない

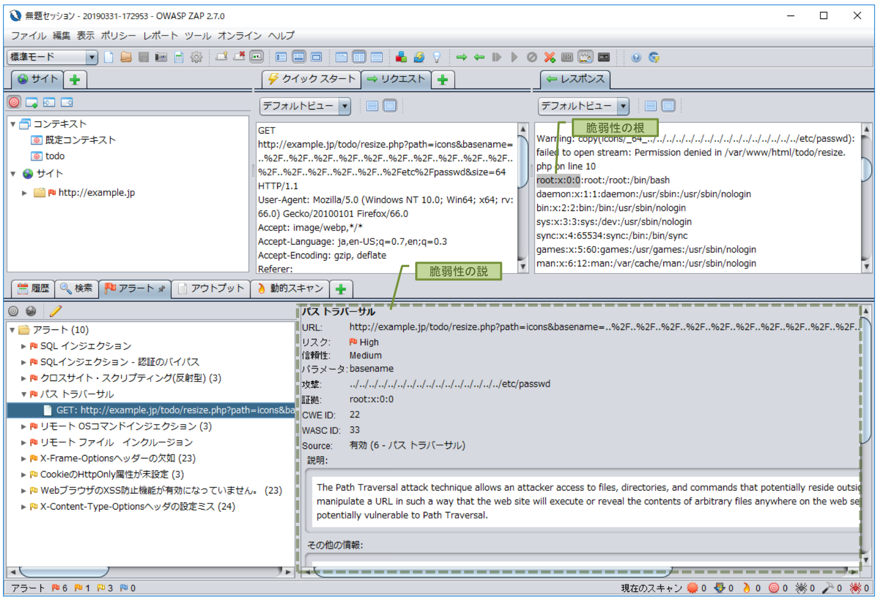


脆弱性診断が開始され、下部の一覧上に
順次脆弱性動作のリクエストが行われて、
表示が流れていく

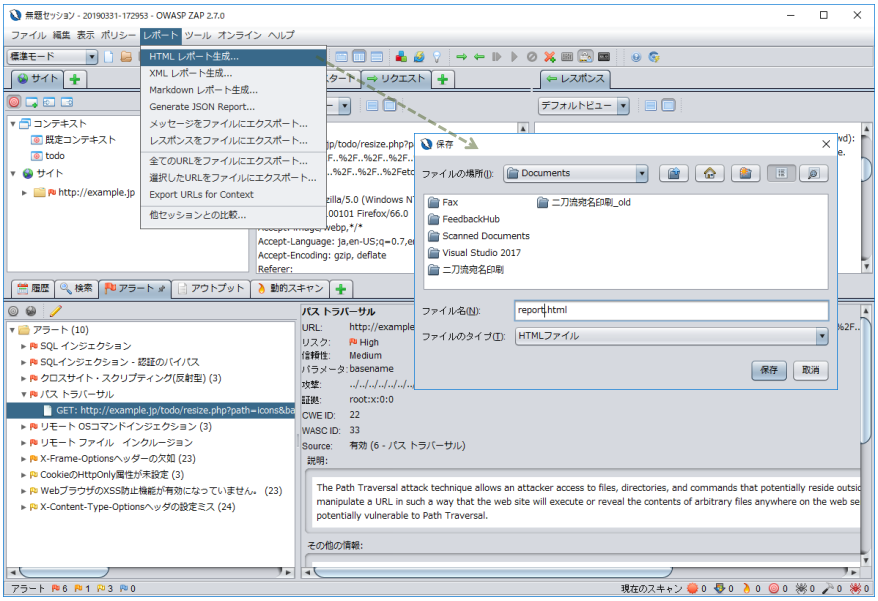


「アラート」タブをクリックすると、脆弱性一覧が表示されます。

パストラバーサルのツリーを展開し、脆弱性情報を表示します。



脆弱性情報のレポートを出力する。

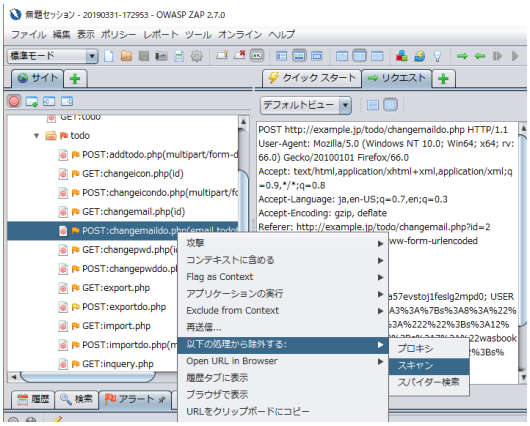


診断の後始末

診断用に作成したアカウントは削除します。脆弱性修正後の動作確認をするために、アカウントを残すこともあります。
診断の結果、データ追加によって、多くのごみが残る場合があります。これらを終了後の後始末として、削除します。

脆弱性診断によって、データが削除、破壊される場合もあるので、できるだけ診断用環境を用意するのがお勧め。
更新系機能は自動診断から除外した上で、更新系機能の診断は手動診断で実施するのがお勧め。

診断から特定のURLを除外



一覧にあるURLをスキャンの対象外にする