

Wikiprint Book

Title: 粒度が細かいパーミッション

Subject: SilverFrost - TracFineGrainedPermissions

Version: 3

Date: 06/04/26 03:30:49

SilverFrost 目次

粒度が細かいパーミッション	3
パーミッションポリシー	3
AuthzPolicy	3
設定方法	3
使用方法	4
利用不可となる機能	6
AuthzSourcePolicy (mod_authz_svn のようなパーミッションポリシー)	6
Trac の設定	6
Subversion の設定	7
デバッグ用パーミッション	7

粒度が細かいパーミッション

Trac 0.11 より前は、リポジトリブラウザ サブシステムだけで「粒度が細かいパーミッション (fine grained permissions)」を定義することができました。

0.11 以降、カスタマイズした パーミッションポリシーのプラグイン を各所に使用するための共通のメカニズムが導入されたので、すべての種類の Trac リソースのあらゆるアクションについて、そのリソースの特定バージョンのレベルまで含めて許可/拒否を設定できるようになりました。

Note: Trac 0.12 では、`authz_policy` はオプションモジュールとして実装されました(`tracopt.perm.authz_policy.*` 配下)。したがって、デフォルトでインストールされ、Trac の管理 Web インタフェースの プラグイン パネルで簡単に有効にすることができます。

パーミッションポリシー

様々なパーミッションポリシーを実装することができます。Trac にはいくつかの例を同梱しています。

現在有効なポリシーは [TracIni](#) の中で設定されているコンフィグレーションによって決定します: 例

```
[trac]
permission_policies = AuthzSourcePolicy, DefaultPermissionPolicy, LegacyAttachmentPolicy
```

このリストの1番目の [#AuthzSourcePolicy](#) ポリシーについては、下に記載しています。続く `DefaultPermissionPolicy` では、[TracPermissions](#) に記載されている従来型の粒度が粗いパーミッションチェックを行いません。そして3番目の `LegacyAttachmentPolicy` は添付ファイルに対して、粒度の粗いパーミッションチェックを行いません。

使用可能なオプションの選択肢として、`Authz` 形式のシステムにてとても一般的なパーミッションポリシーを提供する [#AuthzPolicy](#) があります。詳細については、[authz_policy.py](#) を参照して下さい。

もう一つの評判のよいパーミッションポリシーである、[#AuthzSourcePolicy](#) は pre-0.12 で再実装され、新しいシステムでは、`Subversion` のリポジトリに限定して粒度の細かいパーミッション設定をサポートするようになりました。

その他の例については、[sample-plugins/permissions](#) を参照して下さい。

AuthzPolicy

設定方法

- [ConfigObj](#) をインストールする (0.12 でも必要)
- `authz_policy.py` を `plugins` にコピーする (Trac 0.11でのみ必要)
- [authzpolicy.conf](#) ファイルを適当な場所 (望ましくは、Web サーバ起動ユーザ以外が読み取りできないセキュアな領域) に配置する。ファイルに非ASCII文字が含まれる場合は UTF-8 で保存してください

`trac.ini` ファイルをアップデートする:

`[trac]` セクションの [permission_policies](#) を編集する

```
[trac]
...
permission_policies = AuthzPolicy, DefaultPermissionPolicy, LegacyAttachmentPolicy
```

新規に `[authz_policy]` セクションを追加する

```
[authz_policy]
authz_file = /some/trac/env/conf/authzpolicy.conf
```

[WebAdmin](#)でプラグインを有効にするか、`[components]` のセクションを編集する

```
[components]
...
# Trac 0.12
tracopt.perm.authz_policy.* = enabled
# for Trac 0.11 use this
#authz_policy.* = enabled
```

使用方法

パーミッションポリシーを指定する順序はとても重要です。ポリシーは設定された順序で評価されます。

個々のポリシーはパーミッションチェックに対して True, False, None を返します。ポリシーが明示的にパーミッションを許可する場合は、 True を返します。明示的に拒否する場合は、 False を返します。そして、パーミッションを許可も拒否もできない場合、 None が返されます。

Note: 戻り値が None の場合のみ、 次のパーミッションポリシーに問い合わせを行います。どのポリシーも明示的にパーミッションを許可しない場合、最終的な結果は False となります (つまり、権限なしとみなされます)。

authzpolicy.conf は .ini スタイルの設定ファイルです:

```
[wiki:PrivatePage@*]
john = WIKI_VIEW, !WIKI_MODIFY
jack = WIKI_VIEW
* =
```

- config ファイルの各セクションは Trac のリソース記述子との照合に用いる グローバルなパターンです。記述子は以下のような形式です:

```
<realm>:<id>@<version>[/<realm>:<id>@<version> ...]
```

リソースを親から子の順に、左から右へ記述します。不特定の コンポーネントがある場合は、 * で置き換えられます。バージョンのパターンが明示的に記されていないければ、すべてのバージョン (@*) が暗黙的に追加されます。

例: [WikiStart](#) ページを照合する

```
[wiki:*]
[wiki:WikiStart*]
[wiki:WikiStart@*]
[wiki:WikiStart]
```

例: [WikiStart](#) の 添付ファイル wiki:WikiStart@117/attachment/FOO.JPG@* を照合する

```
[wiki:*]
[wiki:WikiStart*]
[wiki:WikiStart@*]
[wiki:WikiStart@*/attachment/*]
[wiki:WikiStart@117/attachment/FOO.JPG]
```

- セクションは設定ファイルに書かれている 順に 現在の Trac のリソース記述子に対して チェックされます。順番は重要です
- 一度 セッションにマッチすれば、順に 現在のユーザ名がセッションの キー (ユーザ名) と照合されます
- キー (ユーザ名) の前に @ を付けると、グループとして処理されます
- 値 (パーミッション) の前に ! を付けると、そのパーミッションは 拒否されます

通常の Trac パーミッションのルールを適用していれば、ユーザ名は、 'anonymous', 'authenticated', <username> '*' 等とマッチするはずですが。

Note: ユーザによって作成された (例えば、ブラウザ上から 管理 / 権限 (英語版では Admin / Permissions) の '権限グループの追加' (英語版では 'adding subjects to groups')) グループには使えません。詳細については [#5648](#) を参照してください。

例えば、 authz_file が次の内容を含みます:

```
[wiki:WikiStart@*]
* = WIKI_VIEW

[wiki:PrivatePage@*]
john = WIKI_VIEW
* = !WIKI_VIEW
```

デフォルトパーミッションが次のような内容の場合:


```
[trac]
authz_file = /path/to/svnaccessfile
```

authz_file 内でシンタックス [modulename:/some/path] を使用する場合、以下の設定を追加してください:

```
authz_module_name = modulename
```

modulename には、[trac] セクション中の repository_dir に設定したリポジトリと同じものを設定します。例えば [trac] セクション内の repository_dir に /srv/active/svn/blahblah を設定している場合は次のように設定します:

```
[trac]
authz_file = /path/to/svnaccessfile
authz_module_name = blahblah
...
repository_dir = /srv/active/svn/blahblah
```

Subversion の Authz ファイル /path/to/svnaccessfile では、[blahblah:/some/path] のようにエントリを記載します。

Note: Authz ファイルで使用するユーザ名と、Trac で使用するユーザ名は 同じでなければなりません。

0.12 では、trac.ini の permission_policies に AuthzSourcePolicy を必ず含めて下さい。さもないと、authz のパーミッションファイルは無視されます。

```
[trac]
permission_policies = AuthzSourcePolicy, DefaultPermissionPolicy, LegacyAttachmentPolicy
```

Subversion の設定

通常は同じアクセスファイルを対応する Subversion リポジトリに適用します。Apache のディレクティブには以下のように設定してください:

```
<Location /repos>
  DAV svn
  SVNParentPath /usr/local/svn

  # our access control policy
  AuthzSVNAccessFile /path/to/svnaccessfile
</Location>
```

複数のプロジェクト Environment において、プロジェクト全体にどのようにアクセス制限を行うかについての情報は、[TracMultipleProjectsSVNAccess](#) を参照してください。

デバッグ用パーミッション

trac.ini の設定:

```
[logging]
log_file = trac.log
log_level = DEBUG
log_type = file
```

ウォッチコマンド:

```
tail -n 0 -f log/trac.log | egrep '\[perm\]|\[authz_policy\]'
```

どんなチェックが行なわれているか見ることができます。より詳細な情報については、プラグインのソースに添付されているドキュメントを参照して下さい。

See also: [TracPermissions](#), [TracHacks:FineGrainedPageAuthzEditorPlugin](#) は設定を編集するプラグインです。