CORSO DI SICUREZZA DEI SISTEMI WEB E MOBILI

Modulo 4: Sicurezza dei Sistemi Web

Apache: Direttive user-based e Autenticazione

1 Autenticazione

Per l'autenticazione in Apache sono possibili due possibili modi:

- **Autenticazione Basic**: quella più comunemente utilizzata in quanto supportata da tutti i browser. In questo caso, *login* e *password* vengono spediti in chiaro (codificati in `base64`);
- **Autenticazione Digest**: spedisce *login* e *password* cifrati in MD5. Ritenuta in passato più sicura della Basic, visto i "problemi" dell'MD5, è considerata insicura e quindi viene suggerito di utilizzarla con `mod_ssl` e con la trasmissione cifrata.

Inoltre sono possibili diversi `backend` a cui appoggiarsi per il controllo di *login* e *password*.

I moduli standard (già inclusi) sono:

- `mod_auth_basic`: autenticazione base con file;
- `mod_auth_digest`: autenticazione digest con file.

Altri moduli (trovate la lista completa su: `http://modules.apache.org/`) sono:

- `mod_authn_dbd`: carica i dati degli utenti da autenticare da tabelle SQL;
- `mod_authn_dbm`: carica i dati degli utenti da autenticare da file con formato DBM;
- `mod_authnz_ldap`: supporto di LDAP.

1.1 Direttive relative all'autenticazione

Le seguenti direttive vanno inserite quando è necessario autenticare un utente, quindi vanno utilizzate insieme alle direttive user-based che vedremo nella prossima sezione.

- `AuthType Basic`

  Seleziona il metodo di autenticazione (`Basic` o `Digest`).

- `AuthName ''Area Protetta''`

  Assegna un nome all'area (*Realm*) da proteggere. Quando tenteremo di accedere il browser presenterà una finestra di dialogo per l'inserimento di nome utente e password che riporterà tale dicitura. Lo scopo principale è quello di rendere evidente al navigatore che tipo di informazioni riservate stia richiedendo. Le virgolette doppie si rendono necessarie nel caso l'etichetta, come nell'esempio, si componga di più parole separate da spazi.

- `AuthBasicProvider file`

  Specifica il tipo di sorgente per l'autenticazione. Il valore di default è `file`, in tal caso la clausola è opzionale.

- `AuthUserFile PathName`

  Specifica il file che contiene i nomi degli utenti e delle relative password. Per la creazione di questo file viene fornita l'*utility* da riga di comando `htpasswd` che prenderemo in considerazione in seguito.

- `AuthGroupFile PathName`

  Specifica il file che contiene le informazioni sui gruppi. Da utilizzare solo nel caso si autorizzi l'accesso a gruppi di utenti.

Importante: È consigliabile tenere il file delle password (e dei gruppi) al di fuori del document tree del Web server, altrimenti gli utenti potrebbero visualizzarlo con il browser!

1.2 Direttive user-based

La sintassi delle direttive per l'accesso user-based è la seguente:

- `require valid-user`: si richiede che l'utente sia un qualsiasi utente valido contenuto nel file contenente utenti e password;
- `require user` *utente1 utente2*: solo alcuni specifici utenti elencati;
- `require group` *group1 group2*: solo gli utenti appartenenti al/i gruppo/i elencato/i. In questo caso si deve usare anche la direttiva `AuthGroupFile` per specificare il file in cui sono indicati i gruppi.

1.3 Esempio

Prerequisiti:

- Le direttive vanno inserite nel file di configurazione principale (in genere in una sezione `<Directory>`) o in un file `.htaccess`. Nel secondo caso, deve essere settato almeno `AllowOverride AuthConfig` (anche `AllowOverride All` va bene);
- I moduli `mod_authn_core` e `mod_authz_core` (per la versione 2.4) devono essere stati compilati o caricati del file `apache2.conf` (cosa che avviene già di default, quindi in genere non ci si deve preoccupare).

I passi da seguire sono i seguenti:

1. Creare un file delle password usando l'utility `htpasswd`;
2. Configurare il server in modo che visualizzi una pagina di autenticazione, ovvero inserendo il codice:

   ```
   AuthType Basic
   AuthName Area Protetta
   AuthBasicProvider file
   AuthUserFile /usr/local/httpd/passwd/password
   ```

3. Specificare gli utenti che hanno il permesso ad accedere

   ```
   Require user chiara
   ```

In caso di accesso ad un gruppo si deve creare un file dei gruppi che abbia la sintassi seguente:

```
groupName:  chiara alice bob
```

Combinando il tutto, in caso di gruppi, si ottiene:

```
AuthType Basic
AuthName Area Protetta
AuthBasicProvider file
AuthUserFile /usr/local/httpd/passwd/passwords
AuthGroupFile /usr/local/httpd/passwd/groups
Require group groupName
```

1.4 Usare le directive host-based e user-based insieme

Le direttive non sono mutuamente esclusive e possono essere usate in contemporanea, però si **deve** usare la direttiva `satisfy` che definisce come i due tipi di restrizioni interagiscono.

In particolare:

- `satisfy all`: un accesso è permesso se sia le direttive host-based sia le direttive user-based sono soddisfatte;
- `satisfy any`: un accesso è permesso se almeno una direttiva è soddisfatta.

Per esempio:

```
Require valid-user
Order allow,deny
Allow from 192.168.1
Satisfy Any
```

permette l'accesso ad un utente valido **oppure** ad una richiesta proveniente dalla rete `192.168.1`.

1.5 Utility `htpasswd`

Si utilizza questa utility per creare il file delle password o per inserire nuovi utenti nel file delle password. L'utilizzo avviene da riga di comando nel modo seguente:

- `htpasswd -c nome_file username`

  Crea il file (nel caso esista già lo sovrascrive) `nome_file` e inserisce una nuova entry con identificativo *username*; chiede di immettere la password (due volte).

- `htpasswd /etc/apache/passwd user`

  Si può comportare in due modi:

  - se l'utente non esiste, inserisce il nuovo *username* e chiede di inserire la *password*;
  - se l'utente esiste, cambia la *password* all'utente specificato, chiedendola.

Un esempio di *entry* del file è la seguente:

```
username:XWY5JwrAVBXsQ
```

Importante: va detto che si assume che le password vengano comunicate agli utenti per mezzo di un canale out-of-band.