

Esempi di utilizzo del Tool da linea di comando

Uno strumento che ci è molto utile per consultare le variabili API che costituiscono la configurazione di sistema e degli account è il Tool accessibile da linea di comando. In questa guida ne illustriamo brevemente il funzionamento con alcuni esempi pratici di utilizzo.

Sintassi

La sintassi generica di utilizzo dello strumento è la seguente:

```
tool [opzione] <comando> <oggetto> <parametri>
```

Le *opzioni* permettono un uso avanzato dello strumento consentendo ad esempio di applicare un filtraggio alla ricerca o di accedere ad un server remoto.

I *comandi* definiscono le azioni da effettuare (es: creazione, visualizzazione, cancellazione, ecc).

Gli *oggetti* sono le entità sulle quali le azioni vengono eseguite, possono ad esempio essere gli account, i domini o anche l'intero sistema permettendoci così di visualizzare e modificare la configurazione di sistema.

I *parametri* ci permettono invece di definire meglio le azioni da effettuare specificando ad esempio quale variabile mostrare o quale altra modificare e come.

Particolarmente utili risultano essere i caratteri jolly "*" e "%" grazie ai quali è possibile allargare il campo delle proprie ricerche. Ad esempio:

```
tool display account *@*
```

restituirà l'elenco di tutti gli account definiti sul Server.

In caso si volesse effettuare una ricerca più mirata, restringendo la base a tutti gli alias che finiscono in "rto" si potrebbe invece procedere come segue:

```
tool display account %rto@*
```

ottenendo un output come il seguente:

```
umberto@domain1.it  
roberto@domain2.com
```

Esempi pratici di utilizzo

- Nel caso in cui si volesse assegnare l'accesso di un qualsiasi modulo (nell'esempio WebDav) ad un gruppo ristretto di account potrebbe risultare utile togliere il servizio a tutti per poi andare ad abilitarlo manualmente solo allo specifico gruppo di utenti ai quali deve essere consentito.

```
tool set account *@* u_webdav 0
```

- Sempre riguardo alle politiche d'accesso dei servizi è possibile, grazie all'opzione di filtraggio, ottenere un elenco degli account per i quali un determinato servizio è abilitato (nell'esempio Anti-Spam).

```
tool -filter="u_as = 1" display account *@*
```

- Una delle problematiche che più comunemente portano alla violazione di un sistema per fini di diffusione di spam è la definizione di credenziali deboli per uno o più account. Se vogliamo ad esempio individuare tutti gli account con password uguale all'alias possiamo utilizzare il seguente comando:

```
tool -filter="u_alias = u_password" display account *@*
```

- Si tenga presente che tuttavia questa ricerca non individuerrebbe quegli account con alias multipli uno dei quali uguale alla password. Per questo fine è necessario abilitare [Domini e Account > Politiche > Politiche delle password > Confronta le password con il nome utente e gli alias] ed eseguire il comando:

```
tool check account *@* passpolicy
```

ottenendo così una lista di tutti gli account che violano una qualsiasi delle politiche delle password definite.

- Una importante procedura è quella che permette di aggiornare la struttura dei database da quelli di una versione IceWarp precedentemente installata (nell'esempio 8.0.3) all'attuale.

```
tool upgrade 8.0.3
```

- Possiamo anche avere la necessità di effettuare un backup contenente solo un determinato sottoinsieme di dettagli di configurazione degli account e successivamente ripristinarlo dove desiderato.

Assumiamo ad esempio di voler esportare solo il nome utente e password di tutti gli account del dominio "icewarpdemo.com", possiamo farlo come segue:

```
tool export account *@icewarpdemo.com u_mailbox u_password  
> icewarpdemo.csv
```

in questo modo otteniamo un file CSV che possiamo poi importare come segue:

```
tool import account "domain1.csv"
```

Note

L'utilizzo del tool presenta alcune limitazioni nel caso in cui gli utenti siano definiti su database anziché su file system. Infatti mentre ad alcune variabili API della configurazione degli account corrispondono dei campi del database, per altre (es: *u_comment* e *u_mailboxsize*) questa corrispondenza non è verificata e pertanto alcuni comandi del tool si tradurrebbero in interrogazioni impossibili da portare a termine.

Ad esempio se avessimo gli utenti su database e tentassimo un'interrogazione tool come la seguente:

```
tool -filter="u_mailboxsize < 100000" display account *@*
```

non otterremmo alcun risultato e potremmo rilevarne la ragione nel log degli Errori:

```
SYSTEM [0C48] 17:56:48 DB lib Error executing query:  
Unknown column 'u_mailboxsize' in 'where clause'
```

il comando tool è stato tradotto in un'interrogazione SQL che ha nella clausola WHERE un attributo che in realtà non esiste.