Cosign Multi-Factor Specification

3 Novermber 2005
Draft 3

Wesley Craig & Johanna Craig
cosign@umich.edu

Introduction

We provide a general framework to support multiple authentication factors with cosign. The set of factors the user has satisfied are passed to filters. Filters enforce requirements for authentication and communicate the requirements to the central CGI for UI purposes. We also describe the impact on proxy kerberos tickets, replication, and re-authentication.

Filter Configuration

Filters may be configured with a list of required authentication factors. For Apache:

CosignRequireFactor UMICH.EDU OTP

or:

CosignRequireFactor LEVEL2

would indicate that either (UMICH.EDU & OTP) or just LEVEL2 are required to satisfy the filter's multi-factor authentication criteria. If no factors are listed, then no factor checking occurs and any factor is accepted.

Filters may also be configured with a factor suffix which will be ignored. For example:

CosignIgnoreFactorSuffix -junk

This causes the filter to remove the "-junk" suffix from any server-provided factors before comparison with required factors. For example, if the filter requires the factor "OTP", and ignores the suffix "-junk", and the user authenticates with the factor "OTP-junk", then the filter's authentication factor requirements would be fulfilled.

The filter will populate an environment variable, "COSIGN_FACTOR", with a comma separated list of factors in effect for that particular session. Factor suffixes configured with "CosignIgnoreFactorSuffix" will still be passed through. For example:

COSIGN_FILTER=kerberos,otp-junk

For Apache, CosignRequireFactor is valid in the server config, virtual host, location, and directory contexts. In other filters, these or other contexts may or may not make sense. Filters on platforms that do not inherently support the finer grained contexts may need to implement a similar concept in the filter itself in order to support specific applications.

Query String Changes

See RFC 2396. During registration, the query string has the following syntax:

*register-url***?**[**basic&**]*service***=***cookie*[**;**]**&***referring-url*

For example:

https://weblogin.umich.edu/?cosign-webmail=C53H4FKtDb-bkwszVGJEdG3hbp17fQ-qfYPA3-HdyAyXLUxYHOXwwt8c+0bKOw0rO0OaM0CuW0IjS2B7ZaCdM192yt9eOice5cTH549KC2Odb3kcxizKXdBwwioP;&https://web.mail.umich.edu/?mailbox=INBOX

The registration query string syntax changes to:

*register-url***?**[**basic&**][**factors=***factor1*[**,***factor2*]...**&**]*service***=***cookie*[**;**]**&***referring-url*

where "*factor1,factor2*" are configured in the filter with CosignRequireFactors and interpreted by the cosign CGI as the list of factors to present to the user.

CGI Configuration

The CGI may be configured to use PAM for password verification. The "**pam**" option has the following syntax:

pam *pam-service-name factor form-input-name [options]*

The *pam-service-name* is passed to the pam_start() routine. The *factor* is set if authentication succeeds. The *form-input-name* is the password as posted from the login form.

There are currently two options defined. The **second-factor** option means that this factor is only useful with another factor. It is intended for use in environments where repeated authentication failures may cause the target account to be locked. The **user_unknown=***string* option causes "*string*" to be appended to *factor* if the PAM returns PAM_USER_UNKNOWN. This is designed to work in conjunction with *CosignIgnoreFactorSuffix* (see **Filter Configuration**, above) to support phased rollout of OTP tokens. An example:

```
pam cosign-rsa OTP otp second-factor user_unknown=-junk
```

Several "legacy" factors are defined. The "FRIEND" factor is used when accounts are authenticated with the MySQL-email based system defined in the "CoSign Friend" specification. The "BASIC" factor is used when the cosign CGI is protected by Apache, unless the environment variable COSIGN_FACTOR is set, in which case the value of COSIGN_FACTOR is used instead. If Kerberos is used to authenticate the account, the factor is set to the Kerberos "realm" used.

The syntax of the "**cookie**" option is extend to include a list of factors that must be satisfied for re-authentication. The old syntax was:

cookie *cosign-service-name* **reauth**

The new syntax is:

cookie *cosign-service-name* **reauth** *factor1 ...*

The old functionality is retained by converting those options to:

cookie *cosign-service-name* **reauth** UMICH.EDU

See **Multi-factor Re-authentication** below for more details of how multi-factor authentication and re-authentication interact.

Protocol Changes

The STARTTLS verb changes from:

```
C: STARTTLS
S: 220 Ready to start TLS
```

to:

```
C: STARTTLS version-number
S: 220 Ready to start TLS
S: 221 TLS successfully started, protocol version version-number
```

This allows the server to support both new and old clients, and corrects a protocol synchronization issue that occurs when the STARTTLS command fails for some reason. *Should review how SMTP solves this issue.*

The LOGIN verb changes from:

```
C: LOGIN login_cookie ip principal realm [ "kerberos" ]
```

to:

```
C: LOGIN login_cookie ip principal factor [ "kerberos" ]
```

The keyword "kerberos" is reserved, and behaves as in previous versions. New factors are established through agreement within a cosign community. Examples might include: UMICH.EDU, OTP, OTP-junk.

The CHECK verb changes from:

```
        C: CHECK servicecookie
        S: 231 ip principal realm
or
        C: CHECK logincookie
        S: 232 ip principal realm
to:
        C: CHECK servicecookie / logincookie
        S: 233 ip principal factor1 factor2 ...
```

Filters allow access only when all required factors (see **Filter Configuration**, above) are satisfied. If all required factors are not satisfied, the filter sets a new service cookie and redirects the browser to the registration URL, including all required factors (see **Query String Changes**, above).

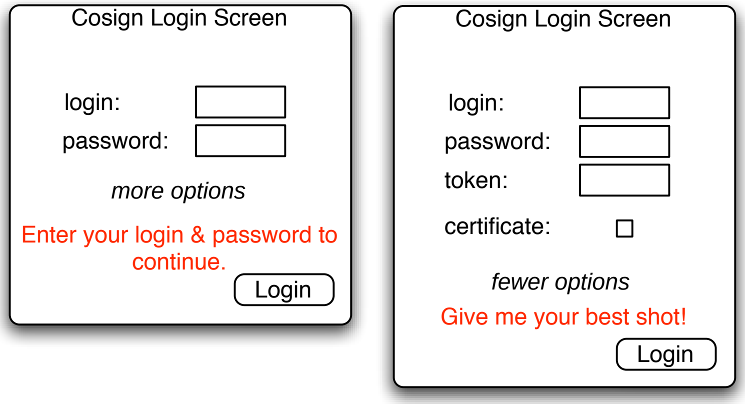
Multi-factor Re-authentication

See the re-authentication specification and **CGI Configuration** above. When a service attempts to register, the re-authentication page will display those factors specified on the query string, minus those factors that have already been satisfied, plus those factors specified in the CGI configuration for that service.

For example, if the user is already authenticated with Kerberos, and attempts to visit a re-authenticating service which is configured with **CosignRequireFactor** to require UMICH.EDU and OTP and with **cookie-reauth** to require just UMICH.EDU, then the user interface will display both UMICH.EDU (as specified by **cookie-reauth**) and OTP (as specified by **CosignRequireFactor**).

User Interface

The user interface presented by cosign is controlled by the filter's configuration, through options provided on the query string. Figure 1 shows the interface when either a service was not visited first or the filter did not specify any authentication factors. The user is permitted to navigate between the two view.

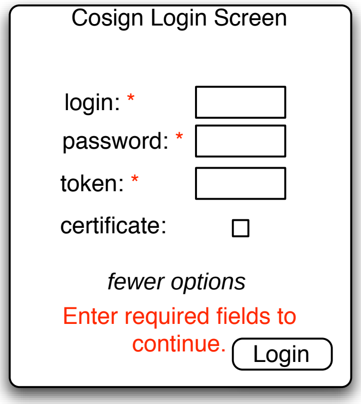


Enter your login & password to continue.

Give me your best shot!

1. Not yet logged in, visiting weblogin.umich.edu first.

Figure 2 shows the interface when the user first interaction with an authenticated service is with a multi-factor protected service. The indicated fields are marked based on factors the filter has included in the query string.



Cosign Login Screen

login: * □

password: * □

token: * □

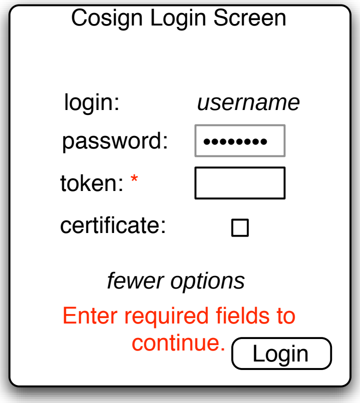
certificate: □

fewer options

Enter required fields to continue.

2. Not yet logged in, visiting multi-factor protected service first.

Figure 3 shows the interface when the user has already authenticated using the UMICH.EDU factor, and then visits a multi-factor protected service. The username and password fields are not editable, and the unsatisfied factor from the query string is indicated.



Cosign Login Screen

login: *username*

password: •••••••••

token: * □

certificate: □

fewer options

Enter required fields to continue.

3. Already logged in, visiting multi-factor protected service.

Pages for re-authentication are similar, with the username never editable.