

サイバーコップスV3.0.0 API仕様書

2007/11/15
株式会社ガーラ

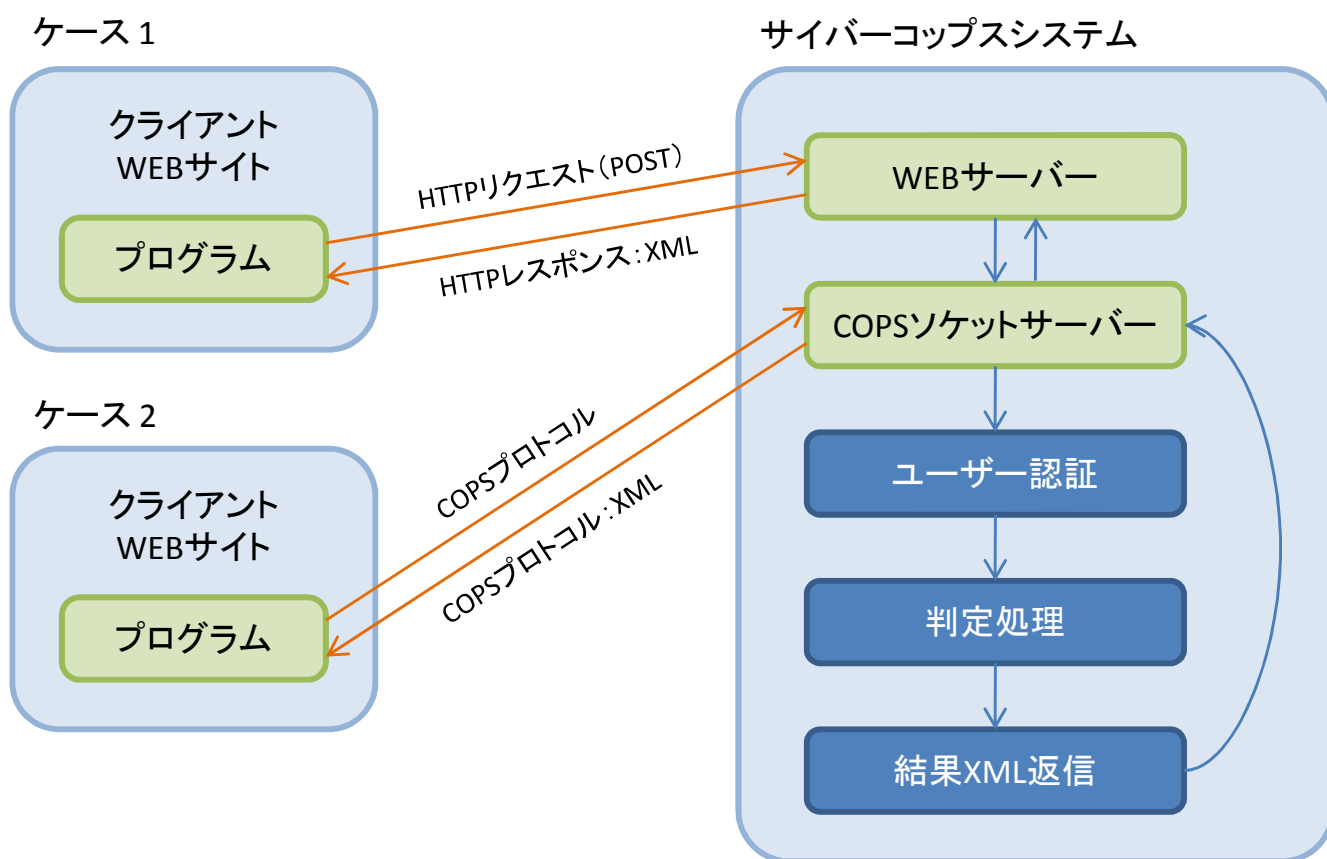
1 システム概要

本システムは、クライアント/サーバー型のアプリケーションである。
クライアント/サーバー間のTCP/IP 通信は、COPS プロトコル、HTTP であり、結果データはXML である。

クライアント(各WEB サイト)は、インターネットを経由して、サイバーコップスシステムに接続し、検閲させたいテキストデータ(投稿文章など)を送信する。

システムは、送信されたテキストデータを形態素解析し、内部で管理している危険単語(フィルターワード)から単語を抽出し、単語単位のレベル付けを行う。

そして、レベル付けされた単語をXML のテキストデータでクライアントに返信するものである。



2 クライアント環境

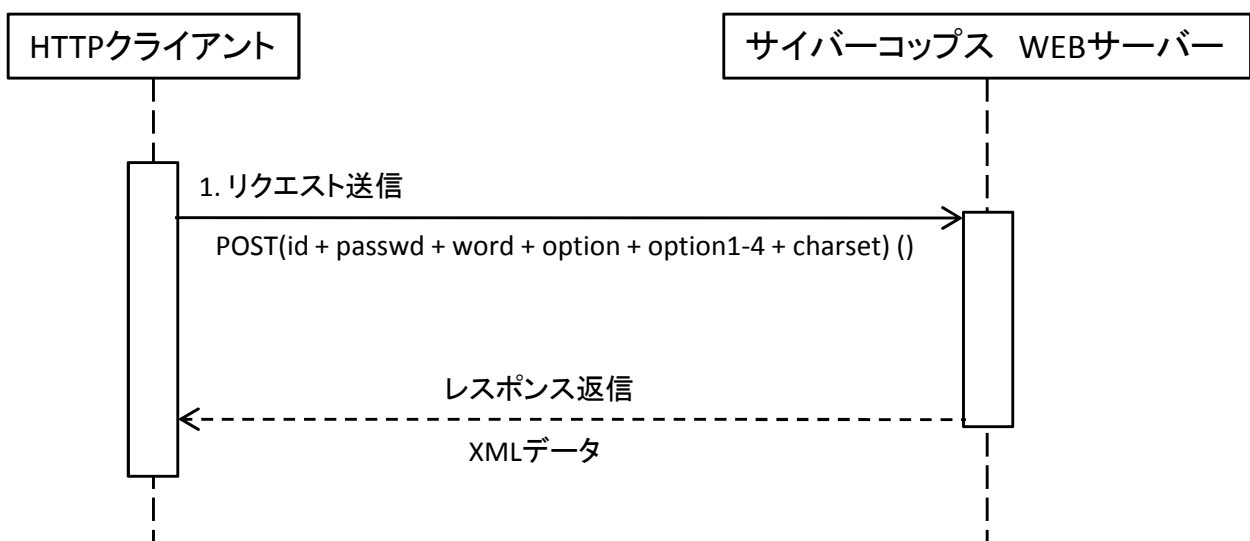
COPS プロトコル、もしくはHTTP 通信できるTCP/IP ネットワークプログラム (WEB アプリ、CGI など) が動作するクライアント。

3 API 仕様

サイバーコップスシステムで提供しているAPIとして2種類がある。
通信プロトコルHTTPを利用したAPIとCOPS 独自通信プロトコル(以下COPS プロトコル)を利用したものである。

3.1 HTTP, HTTPS (HTTP over SSL(TLS))

■ HTTP API を利用した流れ



1. リクエスト送信

HTTP クライアントからリクエストURL「<http://サーバー/cybercops/>」に対して、メソッドPOSTでパラメータid, passwd, word, option, option1, option2, option3, option4, charset をWEB サーバー にリクエストする。

■ HTTPS API を利用した流れ

SSLに対応したHTTPクライアントは、WEBサーバーに接続後、サーバーから証明書(※)が送付され、クライアント側でその証明書の信頼を判断し、暗号化通信を確立する。
暗号化通信が確立された後は、前記のHTTP APIと同じシーケンスで通信が行われる。

■ リクエスト仕様

リクエストURL: `http://COPS サーバー/cybercops/`
`https://COPS サーバー/cybercops/`
最後は”/”でなければならない。

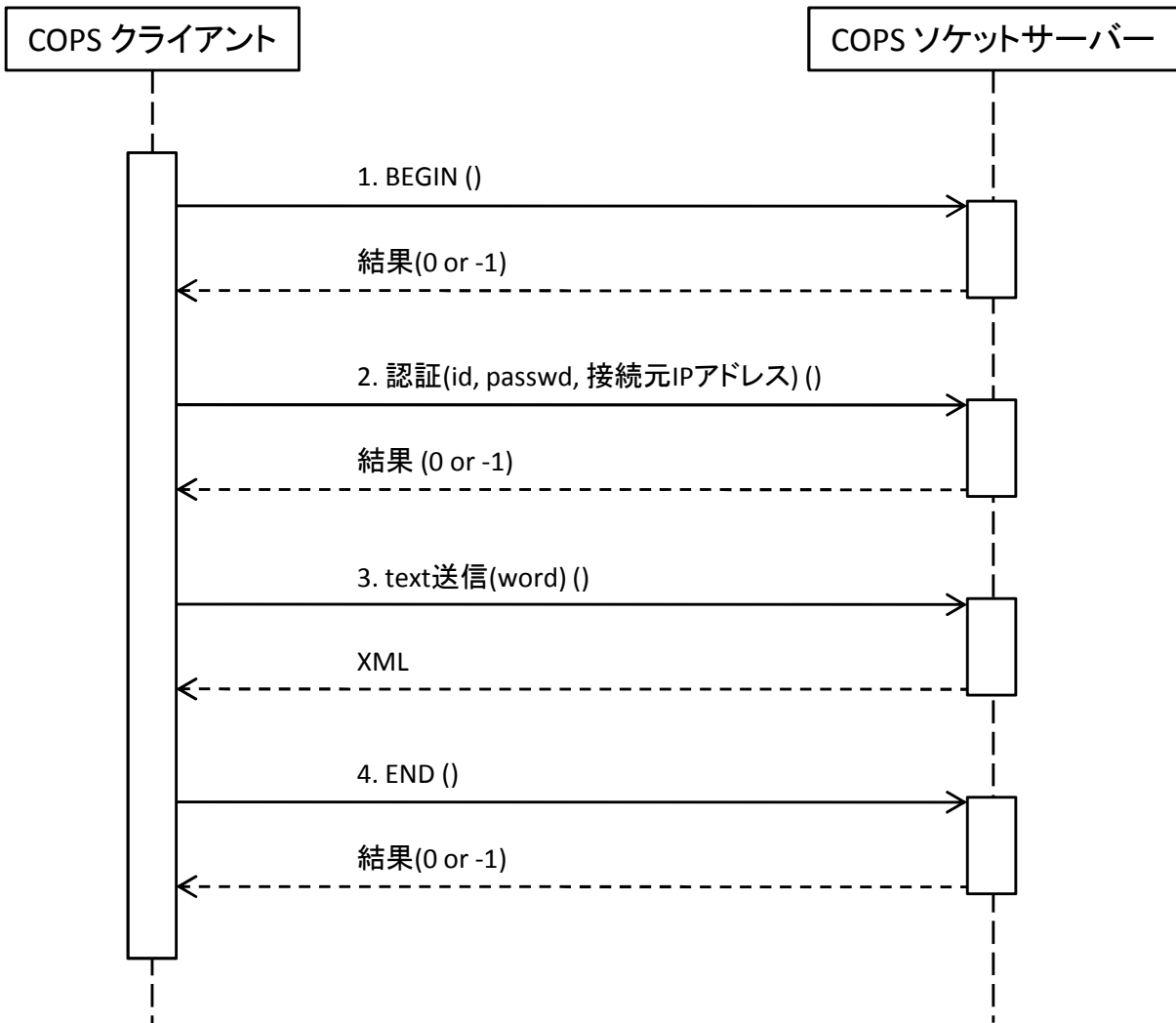
パラメータ	内容
id ※	システム登録されている利用者ID(英数字8文字以下)
ip ※	認証IPアドレス(文字列の形式は、????.????.????.??? (?は10進数) 省略すると、クライアントの接続元IPアドレスを利用する。
passwd ※	利用者IDに対するパスワード(英数字50文字以下)
word ※	テキスト(半角100,000文字以下)
option	テキスト(50文字以下)
option1	テキスト(50文字以下)
option2	テキスト(50文字以下)
option3	テキスト(50文字以下)
option4	テキスト(50文字以下)
charset ※	送信されるテキストの文字コードを指定。指定できるものは、「UTF-8」、「SJIS」、 「EUC-JP」

※ は必須パラメーターである。

リクエストパラメータに日本語等、マルチバイト文字を含む場合、URL エンコードが必要である。

3.2 COPS プロトコル

■ COPS プロトコル API を利用した流れ



※ COPS クライアントは、TCP クライアントであり、COPS サービス (ポート5000 番) へsocket 接続する。

以下の番号は、上図の番号が対応している。

1. COPS クライアントからCOPS サービスに通信開始を知らせる「BEGIN」を送信し、結果を受信する。
結果が正常の場合は0、異常の場合は-1 である。
2. COPS クライアントは、id, passwd 及び接続元IP アドレスをCOPS サービス送信し、ユーザー認証させる。
結果が正常の場合は0、異常の場合は-1 である。
3. 認証に成功後、COPS クライアントは、wordの値をCOPS サービス送信し、COPS 処理(形態素解析、フィルタワードの抽出)を実行させる。
処理された結果のフィルタワード情報をXML で受信する。
4. COPS サービスに通信終了を知らせる「END」を送信し、結果を受信する。
結果が正常の場合は0、異常の場合は-1 である。

■ プロトコル仕様

コマンドは1コマンドに対して以下のフォーマットになる。

[送信バイト数][0x0A (LF)][コマンド]:[データ] [0x0A (LF)]

送信バイト数

コマンド	データ	必須
ID	文字列 (英数字の8文字以下)	○
IP	文字列 接続元IPアドレス (CGIが動作するコンピューターのIPアドレス)を表す文字列 文字列の形式は、????.????.????.??? (?は10進数)となる	○
PASSWD	文字列 (英数字の50文字以下) IDに対するパスワード	○
OPTION	文字列 (50文字以下) コンテンツに関する情報文字列 (ID以外での識別子)	
OPTION1	文字列 (50文字以下) コンテンツに関する情報文字列 (ID以外での識別子)	
OPTION2	文字列 (50文字以下) コンテンツに関する情報文字列 (ID以外での識別子)	
OPTION3	文字列 (50文字以下) コンテンツに関する情報文字列 (ID以外での識別子)	
OPTION4	文字列 (50文字以下) コンテンツに関する情報文字列 (ID以外での識別子)	
WORD	文字列 (100,000文字 (半角) 以下)	○
CHARSET	文字列「UTF-8」,「SJIS」,「EUC-JP」の何れか指定する。 OPTION、OPTION1~4、WORDで渡した文字列の文字コードを指定する。	○
BEGIN	なし。通信処理開始を示す。デリミタはなし	○
END	なし。通信処理終了を示す。デリミタはなし	○

-
- 必須以外は送信しなくても問題ない。
 - コマンドとデータは“:”で繋げる。
ただし、BEGIN 及びEND は、デリミタとデータは必要ない。
 - 送信データの最後には、「LF (0x0A)」を必ず付加する。
 - 「送信データバイト数(半角数字) + LF (0x0A)」を各送信データの先頭に付加する。
データバイト数は、LF も含めた数である。
 - PASSWD コマンドは拡張機能であり、現仕様では使用しないが、送信時には必要である。

■ 通信方法詳細

各コマンドの送信組み合わせと順序と示す。

1. BEGIN を送信

-> BEGIN のみを送信する。

送信バイト数¥nBEGIN¥n

データ送信後、結果を受信し、成否の判定をする。

<- 正常の場合: 0¥n

<- エラーの場合: -1¥n 受信バイト数¥nXML データ¥n

2. ID、IP、PASSWD、OPTION をすべて含めた文字列を送信

-> 各コマンドを別々に送信するのではなく、以下のようにまとめて送信する。

送信バイト数

¥nID:TEST1234¥nIP:192.168.1.1¥nPASSWD:XXXX¥n[OPTION:????????¥n]

データ送信後、結果を受信し、成否の判定をする。

<- 正常の場合: 0¥n

<- エラーの場合: -1¥n 受信バイト数¥nXML データ¥n

3. WORD を送信

-> メッセージを以下のようにWORD コマンドにつなげて送信する。

送信バイト数¥nWORD:メッセージ¥n

データ送信後、結果を受信し、成否の判定をする。

<- 正常の場合: 0¥n

<- エラーの場合: -1¥n 受信バイト数¥nXML データ¥n

4. XML を受信

<- 判定結果の詳細を受信

受信バイト数¥nXML データ¥n

5. END を送信

-> END のみを送信する。

送信バイト数¥nEND¥n

データ送信後、結果を受信し、成否の判定をする。

<- 正常の場合: 0¥n

<- エラーの場合: -1¥n 受信バイト数¥nXML データ¥n

3.3 レスポンス

■ レスポンス仕様

リクエストに対するレスポンスは、XML データである。

文字コードのデフォルトは「UTF-8」であり、WEB 管理ツールから変更できる(SJIS, EUC-JP)。

ただし、エラー時の文字コードは、「UTF-8」固定である。

XML のフォーマットは下記の通りである。

```
<?xml version="1.0" encoding="%s"?>
<result error="%d" words="%d" count="%d" userid="%s" errmsg="%s">
  <word level="%d" count="%d">%s</word>
  <word level="%d" count="%d">%s</word>
  ...
</result>
```

タグ名	内容
result	処理結果を定義するタグ 値: なし 属性: error: COPS エラー番号 word: テキスト中に出現したフィルタワードのユニーク数 count: 文字列中に出現したフィルタワードの総数 userid: COPS 利用者ID(エラー発生タイミングにより、空の場合もある) errmsg: エラー詳細文字列
word	出現したフィルタワードの詳細を定義するタグ (異なるフィルタワードが複数出現した場合は、このタグが複数存在する。) 値: フィルタワード 属性: level: フィルタワードのレベル count: フィルタワードの出現数

例) 正常終了時返却XML(フィルタワード有り)

```
<?xml version="1.0" encoding="UTF-8"?>
<result error="0" words="2" count="5" userid="ogawa" errmsg="">
  <word level="5" count="2">foo</word>
  <word level="9" count="3">bar</word>
  ...
</result>
```

例) 正常終了時返却XML(フィルタワード無し)

```
<?xml version="1.0" encoding="UTF-8"?>
<result error="0" words="2" count="5" userid="ogawa" errmsg="" />
```

例) エラー終了時返却XML

```
<?xml version="1.0" encoding="UTF-8"?>
<result error="105" words="0" count="0" userid="ogawa"
errmsg="USER authentication failed." />
```

■ HTTP APIステータスコード

API の処理結果としての応答ステータスコードは以下のようになる。

コード	説明
200	リクエストに対して、処理が正常終了した場合に返される
500	COPS システム内で問題が発生していた場合に返される
404	不正なURL が指定された場合などに返される

■エラー詳細

XML に設定されるエラー番号とメッセージは、以下である。

エラー	メッセージ	説明
100	Not supported COMANND.	サポートされていないコマンド
101	COMMAND (%s) buffer is overflow. (※ %s はコマンド名)	コマンドの値が制限値を超えている
102	Unknown format.	コマンドのフォーマットが不備
103	COMMAND (%s) is unknown. (※ %s はコマンド名)	知らないコマンド
104	Socket error occurred.	通信に問題が発生した
105	USER authentication failed.	ユーザー認証に失敗
107	Input text size is overflow.	転送されたテキストが100,000 文字を超えた
110	Cops Internal Error(???) (※ ???には複数のメッセージが入る)	COPS 処理エラー
111	Failed to convert ??? string. (※ ???にはコマンド名が入る)	文字コード変換エラー
112	System error occurred.	システムエラー
114	Database timeout. Please Retry!	データベース接続タイムアウトエラー
115	Database error.	データベースエラー
118	[slave service] ignore access from ip.	スレーブとして起動しているので接続が無視された
200	WEBサーバーで発生したエラー	
	Socket error occurred.	通信に問題が発生した
	Couldn't allocate memory.	システムエラー
	Connecting to cops server failed.	COPS サービスが起動していない
	Format Error. ID is empty.	ID が設定されていないか値がない
	Format Error. CHARSET is empty.	CHARSET が設定されていないか値がない
	Format Error. WORD is empty.	WORD が設定されていないか値がない
CHARSET value is invalid. Specify UTF-8, EUC-JP or SJIS.	CHARSET に間違った値が設定されている	