Using reCaptcha with a Python Server

August 19, 2009

Here's how to handle reCaptcha form submissions in Python, on the server side:

```
#!/usr/bin/python

import urllib, urllib2

recaptcha_private_key = '...[your private key goes here]...'

recaptcha_server_name = 'http://www.google.com/recaptcha/api/verify'
recaptcha_server_form = 'https://www.google.com/recaptcha/api/challenge'

def check (client_ip_address, recaptcha_challenge_field, recaptcha_response_field):
    """Return the recaptcha reply for the client's challenge responses"""
    params = urllib.urlencode(dict(privatekey=recaptcha_private_key,
                                   remoteip=client_ip_address,
                                   challenge=recaptcha_challenge_field,
                                   response=to_bytestring(recaptcha_response_field)))
    data = None
    try:
        f = urllib2.urlopen(recaptcha_server_name, params)
        data = f.read()
        f.close()
    except HTTPError:
        pass
    except URLError:
        pass
    return data

def confirm (client_ip_address, recaptcha_challenge_field, recaptcha_response_field):
    """Return True/False based on the recaptcha server's reply"""
    result = False
    reply = check (client_ip_address, recaptcha_challenge_field, recaptcha_response_field)
    if reply:
        if reply.lower().startswith('true'):
            result = True
    return result
```

Just call `confirm (client_ip_address, recaptcha_challenge_field, recaptcha_response_field)` to get the result, either True (i.e., the captcha was completed correctly) or False.

The `client_ip_address`, `recaptcha_challenge_field`, `recaptcha_response_field` fields are provided by parsing the results of POSTing the form (e.g., using a web server module like Mod_python or mod_wsgi).

And the `to_bytestring()` function, which is defined as follows, insures that any unicode character (which do appear occasionally in captchas) get handled correctly:

```
def to_bytestring (s):
    """Convert the given unicode string to a bytestring, using the standard encoding,
    unless it's already a bytestring"""
    if s:
        if isinstance(s, str):
            return s
        else:
            return s.encode('utf-8')
```

This code is now available for download from github.

Update August 3, 2012: Since I'm now learning Go, I've also created go-recaptcha, a package version in that language #golang

---

Archived from the original at http://denis.papathanasiou.org/
₿ Bitcoin Donate: 14TM4ADKJbaGEi8Qr8dh4KfPBQmjTshkZ2