

Forum:[Hack Forums](#)

Thread: [Martingale Betting Script](#)

ExaNet - July 29, 2013, 4:43 pm

Hello, I recently came across a "Martingale" betting script for coinroll.it. But after looking over the code, I came to find that it is not martingale at all. It just chooses a random bet between your min and max. If someone could please edit the following script to make it double your bet if you loose, I will be willing to pay. Just message me your price.

This is the code:

Code `#!/usr/bin/env python`

`# -*- coding: utf-8 -*-`

`#`

`# run.py`

`#`

`# Copyright 2013 [mad]Berry <madberry@hush.com>`

`#`

`# This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.`

`#`

`# This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.`

`#`

`# You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston,
MA 02110-1301, USA.`

`#`

`#`

`### Please send donations to:`

`#btc: 1Q2QDC78uaGgkm4r3G5YzZSriPv7KMUuQU`

`#ltc: Lb5TUyhdUtqsExLmiyqCLAiBZUrs8mEYze`

`# Thanks.`

`import urllib2,urllib,os,simplejson, random`

`# Variables used to run the script`

`user="" #userid from the site`

`password = " #password from the site`

`less_than_min = 32768 #Bet less than "offset"`

`less_than_max = 32768`

`'''`

`64000 = 97.7% odds wins 1.013x`

```
60000 = 91.6% odds wins 1.081x
55000 = 83.9% odds wins 1.179x
32768 = 50.0% odds wins 1.98x
16384 = 25.0% odds wins 3.96x
7000 = 10.7% odds wins 9.268x
2400 = 3.66% odds wins 27.03x
1000 = 1.53% odds wins 64.88x
1 = 0.00153% odds wins 64880x
'''
```

```
max_bet = 200 # Maximum bet 4000=0.00004 || Not used yet
min_bet = 1000 # Minimum bet can't be lower then 1000=0.00001
max_loss = 10000 # stop after X losses
rate_limit = 0 #Rate limit in seconds, currently 1 bet per 10 sec.
```

```
#--Any changes below this line might break the script--
```

```
__prog__ = "coinroll.it automated gambling script"
__scriptname__ = "run.py"
__author__ = "[mad]Berry"
__date__ = "$May 07, 2013 17:10:45 PM$"
__version__ = "1.0.04"
```

```
class Error(Exception):
    pass
```

```
def pause(n, post=False):
    import time, sys
    if post:
        print "Start : %s" % time.ctime()
    while n != 0:
        time.sleep(1)
        sys.stdout.write("Sleeping for: %d \r" % (n) )
        sys.stdout.flush()
        n = n -1
    print " "
```

```
def call(api_url, **kwargs):
    try:
        baseurl = 'https://coinroll.it/'
        api_url = baseurl + api_url
        data = urllib.urlencode(kwargs) #({'user': user, 'password': password})
        header = {
            "Content-type": "application/x-www-form-urlencoded",
            "User-Agent" : "Coinroll Martingale by "+__author__+" Version "+ __version__
        }
        req = urllib2.Request(api_url, data, header)
        response = urllib2.urlopen(req)
        result = response.read()
        result = simplejson.loads(result)
        if 'error' in result:
            # An error occurred; raise an exception
            raise Error({'message': result['error']})
            exit(1)
```

```

return result
except Exception, e:
    print e.args[0]['message']
    exit(1)

```

```

def firstrun():
    print 'Loading first run procedures'
    if os.path.isfile('cred.txt'):
        print 'Success we already have credentials'
        file = open('cred.txt','r')
        address = file.read()
        file.close()
        print address
        pass
    else:
        result = call('deposit/', user=user, password=password)
        file = open('cred.txt','w')
        file.write(result['address'])
        print 'Send a deposit to:', result['address'], 'to create the new account'
        file.close()

```

```

def main():
    resultloss = call('getbalance', user=user, password=password) ##get balance from api
    initial_loss = resultloss['bets'] - resultloss['wins']
    initial_wins = resultloss['wins']
    less_than = random.randint(less_than_min,less_than_max)
    bet = random.randint(min_bet,max_bet)
    while True: #start bet loop
        resultloss = call('getbalance', user=user, password=password)
        losses = resultloss['bets'] - resultloss['wins']
        if (resultloss['balance'] <= 0):
            exit('Reached 300 time to quit')
        elif (resultloss['balance'] >= 400):
            exit('Reached 4500 GOOD time to quit')
        #print resultloss
        if (losses >= max_loss):
            max_lost = initial_loss+max_loss
            #print max_lost
        else:
            max_lost = max_loss
        if (losses == max_lost):
            print 'Lost to much exiting'
            break;
        else:
            result = call('bet',user=user, password=password, lessthan=less_than,amount=bet)
            if (result['win'] == True):
                print "Won! With {2} at {1} \nPlacing another bet I will stop after {0}
losses".format(max_loss,less_than, bet)
            else:
                print "LOST! With {2} at {1} \nPlacing another bet I will stop after {0}
losses".format(max_loss,less_than, bet)
            less_than = random.randint(less_than_min,less_than_max)
            bet = random.randint(min_bet,max_bet)

```

```

        pause(rate_limit)
    return 0

if __name__ == '__main__':
    import logging
    import optparse
    parser = optparse.OptionParser(version=__prog__ + " " + __version__ + "\nDeveloped by: " +
__author__)
    parser.add_option("-n", "--new", action="store_true", dest="new",
        help=("Add a new account, your address will be added to cred.txt"))
    (options, args) = parser.parse_args()
    if options.new == True:
        firstrun()
    else:
        try:
            main()
        except KeyboardInterrupt:
            exit("Bye Bye")

```

ImportUsername - July 29, 2013, 4:48 pm

Code #!/usr/bin/env python

-*- coding: utf-8 -*-

#

run.py

#

Copyright 2013 [mad]Berry <madberry@hush.com>

#

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

#

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

#

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston,
MA 02110-1301, USA.

#

#

Please send donations to:

#btc: 1Q2QDC78uaGgkm4r3G5YzZSriPv7KMUuQU

#ltc: Lb5TUyhdUtqsExLmiyqCLAiBZUrs8mEYze

Thanks.

```
import urllib2,urllib,os,simplejson, random
```

Variables used to run the script

```

user=" #userid from the site
password = " #password from the site
less_than_min = 32768 #Bet less than "offset"
less_than_max = 32768
'''
64000 = 97.7%   odds wins  1.013x
60000 = 91.6%   odds wins  1.081x
55000 = 83.9%   odds wins  1.179x
32768 = 50.0%   odds wins  1.98x
16384 = 25.0%   odds wins  3.96x
7000  = 10.7%   odds wins  9.268x
2400  = 3.66%   odds wins  27.03x
1000  = 1.53%   odds wins  64.88x
1     = 0.00153% odds wins  64880x
'''

max_bet = 200 # Maximum bet 4000=0.00004 || Not used yet
min_bet = 1000 # Minimum bet can't be lower then 1000=0.00001
max_loss = 10000 # stop after X losses
rate_limit = 0 #Rate limit in seconds, currently 1 bet per 10 sec.

```

#--Any changes below this line might break the script--

```

__prog__ = "coinroll.it automated gambling script"
__scriptname__ = "run.py"
__author__ = "[mad]Berry"
__date__ = "$May 07, 2013 17:10:45 PM$"
__version__ = "1.0.04"

```

```

class Error(Exception):
    pass

```

```

def pause(n, post=False):
    import time, sys
    if post:
        print "Start : %s" % time.ctime()
        while n != 0:
            time.sleep(1)
            sys.stdout.write("Sleeping for: %d \r" % (n) )
            sys.stdout.flush()
            n = n -1
        print " "

```

```

def call(api_url, **kwargs):
    try:
        baseurl = 'https://coinroll.it/'
        api_url = baseurl + api_url
        data = urllib.urlencode(kwargs) #({'user': user, 'password': password})
        header = {
            "Content-type": "application/x-www-form-urlencoded",
            "User-Agent" : "Coinroll Martingale by "+__author__+" Version "+ __version__
        }
        req = urllib2.Request(api_url, data, header)
        response = urllib2.urlopen(req)

```

```

result = response.read()
result = simplejson.loads(result)
if 'error' in result:
# An error occurred; raise an exception
    raise Error({'message': result['error']})
    exit(1)
return result
except Exception, e:
print e.args[0]['message']
exit(1)

def firstrun():
    print 'Loading first run procedures'
    if os.path.isfile('cred.txt'):
print 'Success we already have credentials'
file = open('cred.txt','r')
address = file.read()
file.close()
print address
pass
    else:
result = call('deposit/', user=user, password=password)
file = open('cred.txt','w')
file.write(result['address'])
print 'Send a deposit to:', result['address'], 'to create the new account'
file.close()

def main():
    resultloss = call('getbalance', user=user, password=password) ##get balance from api
    initial_loss = resultloss['bets'] - resultloss['wins']
    initial_wins = resultloss['wins']
    less_than = random.randint(less_than_min,less_than_max)
    bet = random.randint(min_bet,max_bet)
    while True: #start bet loop
resultloss = call('getbalance', user=user, password=password)
losses = resultloss['bets'] - resultloss['wins']
if (resultloss['balance'] <= 0):
    exit('Reached 300 time to quit')
elif (resultloss['balance'] >= 400):
    exit('Reached 4500 GOOD time to quit')
#print resultloss
if (losses >= max_loss):
    max_lost = initial_loss+max_loss
#print max_lost
    else:
        max_lost = max_loss
        if (losses == max_lost):
            print 'Lost to much exiting'
            break;
        else:
            result = call('bet',user=user, password=password, lessthan=less_than,amount=bet)
            if (result['win'] == True):
print "Won! With {2} at {1} \nPlacing another bet I will stop after {0}

```

```

losses".format(max_loss,less_than, bet)
    else:
        print "LOST! With {2} at {1} \nPlacing another bet I will stop after {0}
losses".format(max_loss,less_than, bet)
    less_than = random.randint(less_than_min,less_than_max)
    bet = bet*2
    pause(rate_limit)
    return 0

if __name__ == '__main__':
    import logging
    import optparse
    parser = optparse.OptionParser(version= __prog__ + " " + __version__ + "\nDeveloped by: " +
__author__)
    parser.add_option("-n", "--new", action="store_true", dest="new",
        help=("Add a new account, your address will be added to cred.txt"))
    (options, args) = parser.parse_args()
    if options.new == True:
        firstrun()
    else:
        try:
            main()
        except KeyboardInterrupt:
            exit("Bye Bye")
    there you go. That will be 1,000,000 USD

```

ExaNet - July 29, 2013, 4:56 pm

(Jul 29, 2013 04:48 PM)ImportUsername Wrote: [\[CITE\]](#) Code #!/usr/bin/env python

```
# -*- coding: utf-8 -*-
```

```

#
# run.py
#
# Copyright 2013 [mad]Berry <madberry@hush.com>
#
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston,
# MA 02110-1301, USA.
#
#

```

Please send donations to:

#btc: 1Q2QDC78uaGgkm4r3G5YzZSriPv7KMUuQU

```
#ltc: Lb5TUyhdUtqsExLmiyqCLAiBZUrs8mEYze  
# Thanks.
```

```
import urllib2,urllib,os,simplejson, random
```

```
# Variables used to run the script
```

```
user=" #userid from the site  
password = " #password from the site  
less_than_min = 32768 #Bet less than "offset"  
less_than_max = 32768  
""
```

```
64000 = 97.7%   odds wins  1.013x  
60000 = 91.6%   odds wins  1.081x  
55000 = 83.9%   odds wins  1.179x  
32768 = 50.0%   odds wins  1.98x  
16384 = 25.0%   odds wins  3.96x  
7000 = 10.7%   odds wins  9.268x  
2400 = 3.66%   odds wins  27.03x  
1000 = 1.53%   odds wins  64.88x  
1 = 0.00153% odds wins 64880x  
""
```

```
max_bet = 200 # Maximum bet 4000=0.00004 || Not used yet  
min_bet = 1000 # Minimum bet can't be lower then 1000=0.00001  
max_loss = 10000 # stop after X losses  
rate_limit = 0 #Rate limit in seconds, currently 1 bet per 10 sec.
```

```
#--Any changes below this line might break the script--
```

```
__prog__ = "coinroll.it automated gambling script"  
__scriptname__ = "run.py"  
__author__ = "[mad]Berry"  
__date__ = "$May 07, 2013 17:10:45 PM$"  
__version__ = "1.0.04"
```

```
class Error(Exception):  
    pass
```

```
def pause(n, post=False):  
    import time, sys  
    if post:  
        print "Start : %s" % time.ctime()  
        while n != 0:  
            time.sleep(1)  
            sys.stdout.write("Sleeping for: %d \r" % (n) )  
            sys.stdout.flush()  
            n = n -1  
        print " "
```

```
def call(api_url, **kwargs):  
    try:  
        baseurl = 'https://coinroll.it/'  
        api_url = baseurl + api_url
```

```

data = urllib.urlencode(kwargs) #({'user': user, 'password': password})
header = {
"Content-type": "application/x-www-form-urlencoded",
"User-Agent" : "Coinroll Martingale by "+__author__+" Version "+ __version__
}
req = urllib2.Request(api_url, data, header)
response = urllib2.urlopen(req)
result = response.read()
result = simplejson.loads(result)
if 'error' in result:
# An error occurred; raise an exception
raise Error({'message': result['error']})
exit(1)
return result
except Exception, e:
print e.args[0]['message']
exit(1)

```

```

def firstrun():
    print 'Loading first run procedures'
    if os.path.isfile('cred.txt'):
    print 'Success we already have credentials'
    file = open('cred.txt','r')
    address = file.read()
    file.close()
    print address
    pass
    else:
    result = call('deposit/', user=user, password=password)
    file = open('cred.txt','w')
    file.write(result['address'])
    print 'Send a deposit to:', result['address'], 'to create the new account'
    file.close()

```

```

def main():
    resultloss = call('getbalance', user=user, password=password) ##get balance from api
    initial_loss = resultloss['bets'] - resultloss['wins']
    initial_wins = resultloss['wins']
    less_than = random.randint(less_than_min,less_than_max)
    bet = random.randint(min_bet,max_bet)
    while True: #start bet loop
    resultloss = call('getbalance', user=user, password=password)
    losses = resultloss['bets'] - resultloss['wins']
    if (resultloss['balance'] <= 0):
    exit('Reached 300 time to quit')
    elif (resultloss['balance'] >= 400):
    exit('Reached 4500 GOOD time to quit')
    #print resultloss
    if (losses >= max_loss):
    max_lost = initial_loss+max_loss
    #print max_lost
    else:
    max_lost = max_loss

```

```

    if (losses == max_loss):
        print 'Lost to much exiting'
        break;
    else:
        result = call('bet',user=user, password=password, lessthan=less_than,amount=bet)
        if (result['win'] == True):
            print "Won! With {2} at {1} \nPlacing another bet I will stop after {0}
losses".format(max_loss,less_than, bet)
        else:
            print "LOST! With {2} at {1} \nPlacing another bet I will stop after {0}
losses".format(max_loss,less_than, bet)
            less_than = random.randint(less_than_min,less_than_max)
            bet = bet*2
            pause(rate_limit)
            return 0

if __name__ == '__main__':
    import logging
    import optparse
    parser = optparse.OptionParser(version= __prog__ + " " + __version__ + "\nDeveloped by: " +
__author__)
    parser.add_option("-n", "--new", action="store_true", dest="new",
        help=("Add a new account, your address will be added to cred.txt"))
    (options, args) = parser.parse_args()
    if options.new == True:
        firstrun()
    else:
        try:
            main()
        except KeyboardInterrupt:
            exit("Bye Bye")
there you go. That will be 1,000,000 USD
Error:

```

Microsoft Windows [Version 6.1.7600]
Copyright © 2009 Microsoft Corporation. All rights reserved.

C:\Users\User>cd C:\Users\User\Desktop\coinroll_it-master

```

C:\Users\User\Desktop\coinroll_it-master>run.py
File "C:\Users\User\Desktop\coinroll_it-master\run.py", line 119
    bet *= 2
    ^
SyntaxError: invalid syntax

```

```

C:\Users\User\Desktop\coinroll_it-master>run.py
Traceback (most recent call last):
  File "C:\Users\User\Desktop\coinroll_it-master\run.py", line 158, in <
module>
    main()
  File "C:\Users\User\Desktop\coinroll_it-master\run.py", line 119, in m
ain
    bet *= 2
UnboundLocalError: local variable 'bet' referenced before assignment

```

```
C:\Users\User\Desktop\coinroll_it-master>run.py
File "C:\Users\User\Desktop\coinroll_it-master\run.py", line 69
    print "Start : %s" % time.ctime()
    ^
IndentationError: unindent does not match any outer indentation level
```

```
C:\Users\User\Desktop\coinroll_it-master>
```

```
C:\Users\User\Desktop\coinroll_it-master>
```

```
C:\Users\User\Desktop\coinroll_it-master>
```

ImportUsername - July 29, 2013, 4:57 pm

That's just indentations mismatched because of how it got pasted onto the forum. Ill just fix it and put it in pastebin

<http://pastebin.com/LqLjuLAH>

There you go, tell me if that works

ExaNet - July 29, 2013, 5:05 pm

(Jul 29, 2013 04:57 PM)ImportUsername Wrote: [\[CITE\]](#) That's just indentations mismatched because of how it got pasted onto the forum. Ill just fix it and put it in pastebin

<http://pastebin.com/LqLjuLAH>

There you go, tell me if that worksMicrosoft Windows [Version 6.1.7600]
Copyright © 2009 Microsoft Corporation. All rights reserved.

```
C:\Users\User>cd C:\Users\User\Desktop\coinroll_it-master
```

```
C:\Users\User\Desktop\coinroll_it-master>run.py
File "C:\Users\User\Desktop\coinroll_it-master\run.py", line 32
    import urllib2,urllib,os,simplejson, random
    ^
```

```
IndentationError: unexpected indent
Nevermind the post above. Didn't get raw paste data.
Getting this now:
```

Microsoft Windows [Version 6.1.7600]
Copyright © 2009 Microsoft Corporation. All rights reserved.

```
C:\Users\User>cd C:\Users\User\Desktop\coinroll_it-master
```

```
C:\Users\User\Desktop\coinroll_it-master>run.py
Traceback (most recent call last):
  File "C:\Users\User\Desktop\coinroll_it-master\run.py", line 158, in <
module>
    main()
  File "C:\Users\User\Desktop\coinroll_it-master\run.py", line 119, in m
ain
    bet = random.randint(min_bet,max_bet)
  File "C:\Python27\lib\random.py", line 241, in randint
    return self.randrange(a, b+1)
```

File "C:\Python27\lib\random.py", line 217, in randrange
raise ValueError, "empty range for randrange() (%d,%d, %d)" % (istart, istop
, width)
ValueError: empty range for randrange() (1000,201, -799)

C:\Users\User\Desktop\coinroll_it-master>

ImportUsername - July 29, 2013, 5:16 pm

Lets switch this to PM's, I'll PM you a new link

ExaNet - July 29, 2013, 5:18 pm

(Jul 29, 2013 05:16 PM)ImportUsername Wrote: [\[CITE\]](#) Lets switch this to PM's, I'll PM you a new link
Alright sounds good. Thanks, I really appreciate you doing this.