

```

import socket
import SocketServer
import SimpleHTTPServer
import urllib2
import urlparse
import fff
PORT = 54321

class TCPHandler(SocketServer.BaseRequestHandler):
    def handle(self):
        """
        1 character of E
        50 characters of Domain
        10 characters of port
        1 character of N
        GET/POST
        1 character of space
        URL
        1 char of space
        HTTP/1.1
        0d0a
        """
        data = self.request.recv(65536).strip()
        data = fff.dcode(data)
        idx = data.index("\n")
        headers = data[idx+1:]
        domain = data[1:51].strip()
        port = data[51:61].strip()
        data = data[62:].strip()
        idx = data.index(' ')
        operation = data[0:idx].strip()
        data = data[idx+1:]
        idx = data.index(' ')
        url = data[0:idx].strip()

        print "Domain",domain
        print "port ",port
        print "Op ",operation
        print "Url",url
        self.headers = None

        self.path = "http://" + domain + ":" + port + url

        print "in Handle",self.path
        print "Headers ", headers
        try:
            if self.headers == None:
                req = urllib2.Request(self.path)
            else:
                req = urllib2.Request(self.path,None,headers)
            try:
                buff = urllib2.urlopen(req).read()
                print "Dumping this back",buff
            except:
                buff = '''<!DOCTYPE HTML PUBLIC "-//IETF/DTD HTML
2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>

```

```
<h1>Not Found</h1>
<p>The requested URL was not found on this server.</p>
<hr>
<address>Blah </address>
</body></html>'''
                                self.request.sendall(buff)
        except Exception,e:
            print "Out Error",e

httpd = SocketServer.ForkingTCPServer(('',PORT),TCPHandler)
print "serving at %d port " % PORT
httpd.serve_forever()
```