# [Draft] Captive Portal Howto

Prepared by: David Baird at Ideum Inc.
Release date: 20120514
Version 1.0
Site: http://openexhibits.org/research/heist/

*Disclaimer: This document is in no way meant to be a comprehensive guide. It represents notes regarding a proof-of-concept that we are no longer pursuing. We hope this helps you in your own pursuit of a captive portal, but we cannot offer help or support on this topic.*



## History

(2011-Dec-01 David Baird) Updated Python script to fake http://www.msftncsi.com/ncsi.txt (used in Windows 7 to detect internet connectivity).

# Ideum blackhole captive portal 1

## Overview

This captive portal was implemented circa October 2011. The captive portal works by implementing the following functions:

1. Firewall: Rewrite firewall to intercept all DNS packets (by listening to port 53 and redirecting to local DNS server).
2. DNS: Return our own IP address as a response to *any* DNS request, even addresses that do not exist on other networks.
3. HTTP server: Use virtual hosts, and clever web-based scripts, to emulate “the internet.” Special hacks have to be put in place for Apple devices. (TODO: what is the problem with iOS 5.0.1?).
    a. Serve success.html for Apple devices.
    b. Serve www.openexhibits.org
    c. Serve heist-portal.openexhibits.org (running the node.js Heist service)
    d. Redirect all other requests to www.openexhibits.org
4. Wifi Access Point: Get a card capable of master mode or access point mode (TODO: insert links that have more information), and setup hostapd.
5. DHCP: Automatically assign wifi users IP addresses.

TODO: Draw block diagram of processes and interactions.

## Setting up prerequisite packages

Install some packages for Ubuntu:

```
apt-get update
apt-get install vim
apt-get install samba # if you want WINS support on eth0
apt-get install dnsmasq
apt-get install bind9
apt-get install hostapd
apt-get install libapache2-mod-wsgi
#apt-get install nodejs
apt-get install git-core curl build-essential openssl libssl-dev
```

Install the node.js webserver and some packages for it:

```
git clone https://github.com/joyent/node.git && cd node
./configure
make
sudo make install
node -v
cd
curl http://npmjs.org/install.sh | sudo sh
npm install formidable
npm install node-uuid
npm install choreographer
```

## Firewall configuration

/etc/network/interfaces

```
auto lo
iface lo inet loopback
auto eth0
iface eth0 inet dhcp
auto eth1
iface eth1 inet dhcp
iface wlan0 inet manual
iface mon.wlan0 inet manual
```

Firewall rules to intercept and redirect DNS to self, and to allow limited access to local services:

```
ifconfig wlan0 10.7.7.1 netmask 255.255.255.0
iptables -P INPUT ACCEPT
iptables -F
iptables -t mangle -F
iptables -t nat -F
iptables -t nat -A PREROUTING -i wlan0 -p tcp --dport 80 -j REDIRECT
iptables -t nat -A PREROUTING -i wlan0 -p udp --dport 53 -j REDIRECT
iptables -t nat -A PREROUTING -i wlan0 -p tcp --dport 53 -j REDIRECT
iptables -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
iptables -A INPUT -i lo -j ACCEPT
iptables -A INPUT -i eth0  -p tcp --dport  22 -j ACCEPT # SSH
iptables -A INPUT -i eth0  -p tcp --dport  42 -j ACCEPT # WINS
iptables -A INPUT -i eth0  -p tcp --dport 137 -j ACCEPT # WINS
iptables -A INPUT -i eth0  -p udp --dport 137 -j ACCEPT # WINS
iptables -A INPUT -i eth1  -p tcp --dport  22 -j ACCEPT # SSH
iptables -A INPUT -i eth1  -p tcp --dport  42 -j ACCEPT # WINS
iptables -A INPUT -i eth1  -p tcp --dport 137 -j ACCEPT # WINS
iptables -A INPUT -i eth1  -p udp --dport 137 -j ACCEPT # WINS
iptables -A INPUT -i wlan0 -p tcp --dport  80 -j ACCEPT # HTTP
iptables -A INPUT -i wlan0 -p udp --dport  53 -j ACCEPT # DNS
iptables -A INPUT -i wlan0 -p tcp --dport  53 -j ACCEPT # DNS
iptables -A INPUT -i wlan0 -j ACCEPT
iptables -P INPUT DROP
```

## DNS Configuration

/etc/bind/named.conf:

```
options {
  //directory "/etc/bind";
  //pid-file  "/var/run/named/pid";
  directory "/var/cache/bind";
  allow-query { any; };
  allow-recursion { any; };
};

zone "." {
  type master;
  file "/etc/bind/db.catchall";
};
```

/etc/bind/db.catchall:

```
$TTL  604800
@     IN    SOA   . root.localhost. (
                            1          ; Serial
                       604800          ; Refresh
                        86400          ; Retry
                      2419200          ; Expire
                       604800 )        ; Negative Cache TTL

      IN    NS    .
.     IN    A     10.7.7.1
*.    IN    A     10.7.7.1
```

## HTTP Server Configuration

Prepare Apache:

```
#a2enmod rewrite
a2enmod proxy
a2enmod proxy_http
mkdir /var/www/openexhibits.org
mkdir /var/www/heist-portal.openexhibits.org
```

(This is a hack, I know... improvements are welcome) /etc/apache2/sites-available/default:

```
<VirtualHost *:80>
	DocumentRoot /var/www
	<Directory />
	Options FollowSymLinks
	AllowOverride None
	</Directory>
	<Directory /var/www>
	Order allow,deny
	Allow from all
	</Directory>
	WSGIScriptAlias / /var/www/portal.py
	<Directory /root/htdocs>
	Order allow,deny
	Allow from all
	</Directory>
</VirtualHost>

<VirtualHost *:80>
	ServerName www.openexhibits.org
	ServerAlias openexhibits.org
	DocumentRoot /var/www/openexhibits.org
	<Directory />
	Options FollowSymLinks
	AllowOverride None
	</Directory>
	<Directory /var/www/openexhibits.org>
	Options FollowSymLinks MultiViews
	Order allow,deny
	Allow from all
	</Directory>
</VirtualHost>

<VirtualHost *:80>
	ServerName heist-portal.openexhibits.org
	ProxyPreserveHost On
	ProxyVia full
	<proxy>
	Order deny,allow
	Allow from all
	</proxy>
	#RewriteEngine on
```

```
        #RewriteRule ^/(.+) http://localhost:9980/$1 [P]
        ProxyPass / http://localhost:9980/
        ProxyPassReverse / http://localhost:9980/
    </VirtualHost>
```

/var/www/portal.py

```
    def application(environ, start_response):
        if False: pass
        elif environ.get('HTTP_HOST', '').lower().endswith('.apple.com') and \
            environ.get('PATH_INFO', '').lower() == '/library/test/success.html':
            status = '200 OK'
            output = open('/var/www/apple-success.html').read()
            response_headers = [
                    ('Content-type', 'text/html'),
                    ('Content-Length', str(len(output))),
            ]
            start_response(status, response_headers)
            return [output]
        elif environ.get('HTTP_HOST', '').lower().endswith('.msftncsi.com') and \
            environ.get('PATH_INFO', '').lower() == '/ncsi.txt':
            status = '304 Not Modified'
            output = 'Microsoft NCSI'
            response_headers = [
                    ('Content-Length', str(len(output)))
            ]
            start_response(status, response_headers)
            return [output]
        else:
            status = '302 Found'
            output = ''
            response_headers = [
                    ('Content-type', 'text/plain'),
                    ('Content-Length', str(len(output))),
                    ('Location', 'http://openexhibits.org/konnectus/index.html')
            ]
            start_response(status, response_headers)
            return [output]
```

## Wifi access point

TODO: Links for how to find hardware which supports access point mode.
On startup (/etc/rc.local), run this command to start hostapd:

```
cat > /tmp/myhostapd.conf << EOF
interface=wlan0
driver=nl80211
ssid=heist
channel=1
EOF
hostapd -B /tmp/myhostapd.conf
```

## DHCP configuration

On startup (/etc/rc.local), run this command to start dnsmasq:

```
killall dnsmasq
cat > /tmp/mydnsmasq.conf << EOF
interface=wlan0
dhcp-range=10.7.7.50,10.7.7.250,12h
```

```
#dhcp-option=option:router,10.7.7.1
EOF
# -p 0 == disable DNS operation
dnsmasq -p 0 -C /tmp/mydnsmasq.conf
```

## Running the node.js server

```
echo >> /root/heist.log
date >> /root/heist.log
#nohup /usr/local/bin/node /root/main.js >> /root/heist.log &
nohup /usr/local/bin/node /root/main.js > /dev/null &
```

## Random other nice things

Setup WINS services (to find the portal's IP address more easily, for doing e.g. SSH login).
TODO: post overall init script, and samba WINS configuration.