History of AppleTV hacking

featuring Kevin Bradley a.k.a. @nitoTV a.k.a. |bile|
a.k.a. Lechium

Who Am I?

- Started as a "designer"
- Developing on Mac OS and iOS devices for 8+ years
- Wannabe comedian
- notorious twitter loudmouth
- Apple junkie

How did I start?

- Hypercard!!!! (YES!)
- Applescript Studio Apps
- Books by Aaron Hillegass from Big Nerd Ranch
- Sample code (open source, github, stack overflow)
- trial and error

What I've worked on

- SeasOnPass Mac - AppleTV 2 Jailbreaking tool
- greenpoisOn RC 6.1 - iOS Jailbreaking tool
- nitoTV + Installers - many things to many platforms. (for AppleTV 1 and AppleTV 2)
- AirControl - HTTP control of the AppleTV 2

nito

WWDC

AppleTV 1

- 1 GHz intel processor, 256 MB RAM running Mac OS 10.4.7 custom version
- March 21, 2007 - Shipped AppleTV 1
- March 23, 2007 - Hacked to run different formats through wrappers / scripts

Hacking AppleTV 1 1.0

Needed:
*External USB Drive or ipod (I used an ipod)
*boot.efi from appletv
*licensed copy of os x for intel
*usb hub w/ external power

1. Install full osx onto the ipod, using semthex's kernel and method http://www.appletvhacks.net/2007/04/01/mac-os-x-running-on-apple-tv/
2. Modify the AppleFileSystemDriver
****Very special thanks to Turbo for this patch, without him I might still be trying to get the OS to boot off USB**
Add:

```
<dict>
        <key>Content Hint</key>
        <string>5265636F-7665-11AA-AA11-00306543ECAC</string>
        <key>Leaf</key>
        <true/>
</dict>
<dict>
        <key>Content Hint</key>
        <string>Apple_Recovery</string>
        <key>Leaf</key>
        <true/>
</dict>
```

in: /System/Library/Extensions/AppleFileSystemDriver.kext/Contents/Info.plist in the IOPropertyMatch array

If you dont do this step then OSX wont boot

Now we are going to change the partition info to that of a recovery system (THIS PART VARIES ON EACH SYSTEM)
2. sudo gpt show /dev/disk1
You should get results such as:

```
      start        size  index  contents
          0           1         PMBR
          1           1         Pri GPT header
          2          32         Pri GPT table
         34           6
         40      409600      1  GPT part - C12A7328-F81F-11D2-BA4B-00A0C93EC93B
     409640   116538416      2  GPT part - 48465300-0000-11AA-AA11-00306543ECAC
  116948056      262151
  117210207          32         Sec GPT table
  117210239           1         Sec GPT header
```

index 1 is the EFI partition
index 2 is the osx installation
now do this to set it up to work as recovery partition
THIS ASSUMES /dev/disk1 is your external drive you can break your install if it isnt

Electronista / Something awful

Apple TV hacked to run XviD, other formats

updated 09:20 am EDT, Fri March 23, 2007

f 0 160 di 12

Apple TV Already Hacked

The Apple TV has already been hacked to run non-supported video formats, according to a pair of forum users. Confirming Walt Mossberg's claim that the media hub runs a custom version of Mac OS X, the technique involves removing the hard drive and mounting it on a Mac, where it appears as a native HFS volume. Installing the SSH server Dropbear, the video container Perian, and a custom script lets the Apple TV play files outside of its normal MPEG-4 and H.264 standards.

The solution is not failsafe, the discoverers say: the solution requires the creation of reference QuickTime movies, preventing the hub from directly synchronizing the movies themselves. No video formats outside of the XviD standard have been tested so far, they add. However, the discovery suggests that the Apple device will be relatively easily accessible for hobbyists.

By Electronista Staff

First Plugin POC

Apple TV: Finder Hiding Widget

Monday April 23, 2007 11:39AM
by Erica Sadun in Technical

BOOKMARK

So here it is. The proof of concept Finder Hiding widget. So why did I bother? What's the big idea? It goes basically like this: Apple TV (and, presumably the upcoming iPhone) can run many Intel Mac OS X applications because Apple TV is, essentially, an OS X computer that runs a slight variant on the OS. However, Finder wants to take over and control the way users interact with the unit. It refuses to hide, it refuses to hand over control to another app.

More after the jump...

Sure, you can ssh your way in and replace Finder.app with another application or use other low down dirty tricks to get Finder to behave by, essentially, breaking it but my goal was to expand the way people could interact with Finder without replacing it or breaking the system. In other words, I wanted to prove that you could write a plug-in that adds itself to Finder and allows you to call and execute another application, handing over interactive control to that application until you quit and returned to the Apple TV interface.

The first step to this was my Perl plug-in. It allows you to run arbitrary perl code and returns a simple alert screen showing the output of that program. But the program runs in the background. I wanted to push things further. Sure, I could make VLC start playing back a movie but it played *behind* Finder. You could hear it but not see it and you could not interact with it.

So where are things now? The Hider plug-in tells Finder to close and hide its windows and it stops the screen saver from starting up. The application you run (it's compiled in right now, and it's VLC–which must be ported over by hand to Apple TV) can now be seen and used. Finder, using NSTask, waits until the subprocess terminates, and then returns. It's still really crude–I'd much prefer bringing back the finder by pressing Menu, and having the child task pause until re-selected, and I should store the previous screenSaverTimeout instead of just assigning 20 seconds by fiat, but say it with me: "Proof of Concept", not deliverable product.

If you want to give the Hider appliance a spin, you can download a copy here: Download file. Use at your own risk and please do not sue.

Plugin POC cont...

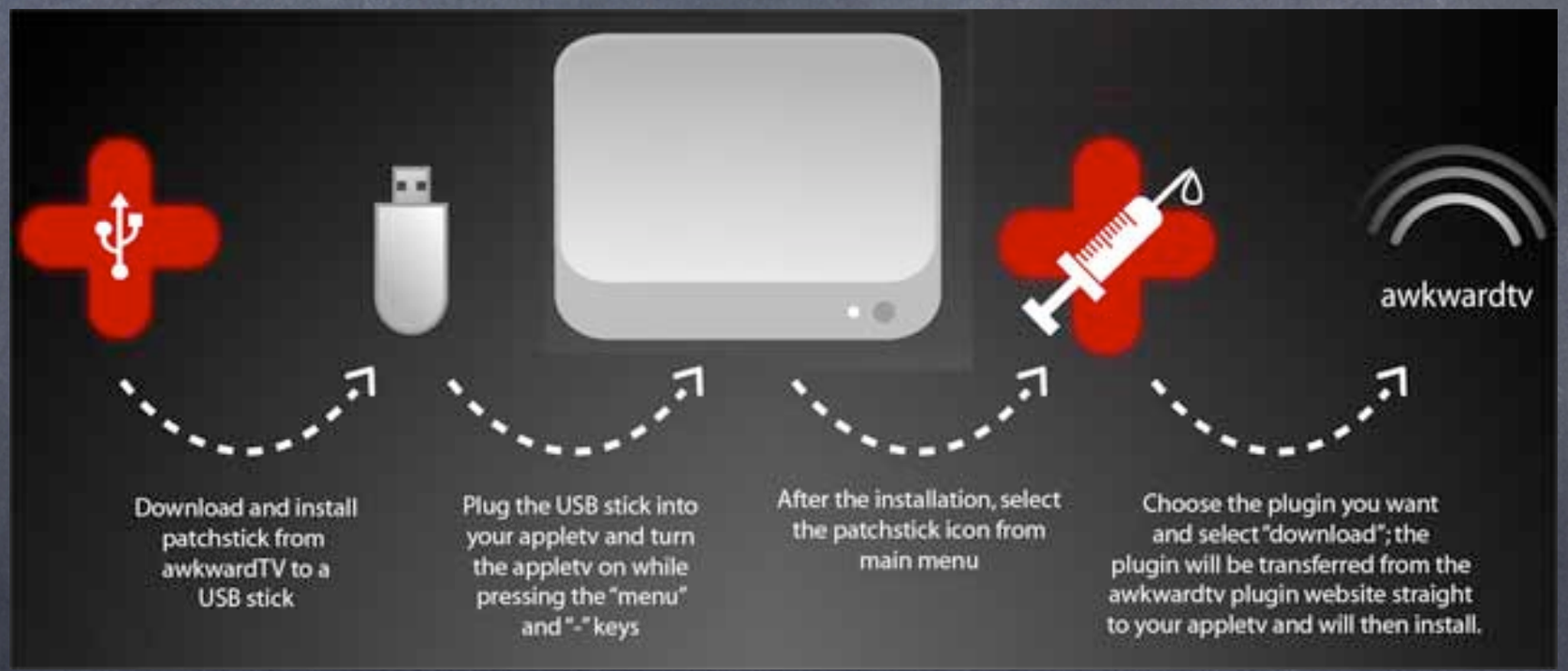
```
- (id)applianceControllerWithScene:(id)scene {

    NSTask *myTask = [[NSTask alloc] init];
    [myTask setLaunchPath:@"/Applications/VLC.app/Contents/MacOS/VLC"];

    [[NSNotificationCenter defaultCenter] postNotificationName: @"BRDisplayManagerStopRenderingNotification"
                                                        object: [BRDisplayManager sharedInstance]];
    [[BRSettingsFacade settingsFacade] setScreenSaverTimeout:0];
    [[BRDisplayManager sharedInstance] releaseAllDisplays];
    [myTask launch];
    [myTask waitUntilExit];
    [[BRDisplayManager sharedInstance] captureAllDisplays];
    [[BRSettingsFacade settingsFacade] setScreenSaverTimeout:20];
    [[NSNotificationCenter defaultCenter] postNotificationName: @"BRDisplayManagerResumeRenderingNotification"
                                                        object: [BRDisplayManager sharedInstance]];
    [scene renderScene];
    return self;
}
```

Patchstick Project

- Old process too convoluted
- High bar for entry



Patchstick 1.0

- Needed OS X 10.4 Image / Disc / Comp
- Idea developed by MacTijn / awkwardTV
- Frameworks, Extensions, Binaries from 10.4.x
- minimal installation of "OS X" needed to hack
- thumb drive of any size should work.

Example Script

```
if [ ! -e /Volumes/OSBoot/System/Library/CoreServices/boot.efi ]; then
        echo "boot.efi on volume 'OSBoot' not found! Have you mounted the 'OSBoot' volume?"
        echo "Exiting.."
        exit 1
fi


partition_disk() {
    echo
    echo " Partitioning $THEDISK:"
    diskutil partitionDisk $THEDISK 2 GPTFormat HFS+ Patchstick-root 80M HFS+ Patchstick 40M
    echo " Patitions done."
    #sleep 2
}

basic_folders() {
    echo "Creating System folders..."
    mkdir /Volumes/Patchstick-root/sbin /Volumes/Patchstick-root/etc /Volumes/Patchstick-root/dev /Volumes/Patchstick-root/
OSBoot /Volumes/Patchstick-root/stuff
    mkdir -p /Volumes/Patchstick-root/usr/lib/system
    mkdir -p /Volumes/Patchstick-root/System/Library/Extensions
    mkdir -p /Volumes/Patchstick-root/System/Library/Frameworks
    ln -s /Volumes/Patchstick-root/sbin /Volumes/Patchstick-root/bin
    mkdir -p /Volumes/Patchstick-root/System/Library/Frameworks/OSXFrames/
    mkdir -p /Volumes/Patchstick-root/usr/libexec

}

find_tiger() {
   version_plist="System/Library/CoreServices/SystemVersion.plist"
   for vol in /Volumes/*; do
      if [ "${vol}" == /Volumes/OSBoot ]; then
         # skip the (hopefully mounted) OSBoot volume
         continue
      fi
```

Example Script Cont...

```
if [ -r "$vol/$version_plist" ]; then
        version=`defaults read "$vol/${version_plist%.plist}" ProductVersion`
        majorVersion="`expr "$version" : "\(10.[0-9]\).*"`"
        if [ xx$majorVersion == xx10.4 ]; then
            echo Using Tiger volume found at "$vol"
            echo -n "ok? [ync?] "
            gotAnswer=0
            while [ $gotAnswer == 0 ]; do
                    read ans
                    case $ans in
                        y) TIGER="$vol"; break 2 ;;
                        n) gotAnswer=1 ;;
                        c) echo "Cancelling.."; exit 0 ;;
                        ?) echo -n "[yes no cancel ?help] " ;;
                    esac
            done
        fi
    done
}

copy_system() {
        echo "Copying System Extentions..."
        cp -R "${TIGER}"/System/Library/Extensions/
{AppleACPIPlatform,AppleAPIC,AppleEFIRuntime,AppleFileSystemDriver,AppleFlashNVRAM,AppleHDA,AppleHPET,AppleIRControll
er,AppleRTC,AppleSMBIOS,AppleSMC,AudioIPCDriver,BootCache,GeForce,IO80211Family,IOACPIFamily,IOATAFamily,IOAudioFa
mily,IOGraphicsFamily,IOHIDFamily,IONDRVSupport,IONetworkingFamily,IOPCIFamily,IOPlatformPluginFamily,IOSCSIArchitectureM
odelFamily,IOStorageFamily,IOUSBFamily,IOUSBMassStorageClass,NVDANV40Hal,NVDAResman,OSvKernDSPLib,System,AppleI
ntelCPUPowerManagement}.kext /Volumes/Patchstick-root/System/Library/Extensions/
        cp -R "${TIGER}"/System/Library/Frameworks/{CoreFoundation,IOKit}.framework /Volumes/Patchstick-root/System/Library/
Frameworks
        cp "${TIGER}"/bin/{bash,chmod,cp,sleep,sync,sh,ls} \
           "${TIGER}"/usr/sbin/{chown,bless} \
           "${TIGER}"/sbin/{mount,mount_hfs,mount_devfs,umount,reboot} \
           /Volumes/Patchstick-root/sbin/
        cp "${TIGER}"/usr/lib/{dyld,{libSystem.B,libncurses.5.4,libgcc_s.1}.dylib} /Volumes/Patchstick-root/usr/lib/
        cp "${TIGER}"/usr/lib/system/libmathCommon.A.dylib /Volumes/Patchstick-root/usr/lib/system/
}
```

Patchstick 2.0

- Linux bootloader by Scott Davilla (XBMC)

- ATV-Bootloader
  - atv-bootloader which uses principals from mach_linux_boot to boot a compiled-in Linux kernel and then finds and boots another Linux kernel using kexec (a user-land kernel bootloader). In addition, atv-bootloader translates several EFI structures into standard PC bios structures. This allows a standard Linux kernel to be booted without the numerous EFI patches required by both mach_linux_boot and mb_boot_tv.
  - atv-bootloader can a) search and find an existing grub menu.lst and auto-boot from it, b) search for a "boot_linux.sh" file and auto-execute it, and c) search for "patchstick.sh" and auto-execute it. The initrd "boot_linux.sh" script currently search sdb then rest of the disk devices (this will be changed in the future once the real boot device can be recovered from the device tree passed by boot.efi). This procedure allows atv-bootloader to auto-boot a linux install or be extended for other uses. This is controlled by a "Kernel Flag" string param in com.apple.Boot.plist and has the following definitions

```
#search for mb_boot_tv, grub and syslinux/ioslinux config --auto boot default
"atv-boot=auto"

#search for a "boot_linux.sh" file and execute it.
"atv_boot=manual"

#search for a "patchstick.sh" file and execute it.
"atv_boot=patchstick"

#default is to drop to a login prompt (user=root, password=root)
```

atv-bootloader is located here and the source is in svn. **Warning SVN trunk is the development branch** You will not find any files named atv-bootloader as atv-bootloader is the concepts contained within the downloaded package. This package contains a prebuilt "fake" mach kernel with embedded Linux kernel/initramfs along with the required support files. The only thing required is boot.efi which cannot be distributed.

Useful plugins / software

- Perian (play other codecs)
- ATVFiles (file browser / media player)
- nitoTV (mplayer frontend / playlist manager / software installer / emulator ... etc)
- awkwardTV loader (install plugins)
- Sapphire (organize, fetch movie / tv show meta, watch movies)

Important People

- MacTijn (awkwardTV founder / patchstick inventor)
- Scott Davilla (major XBMC dev, linux bootloader)
- Turbo (yes the iOS one) - partition work / USB/SSE3 kernel patch/emulation
- Eric Steil III (ATVFiles)
- Graham Booker (perian) Patrick Merrill + gbooker (sapphire)

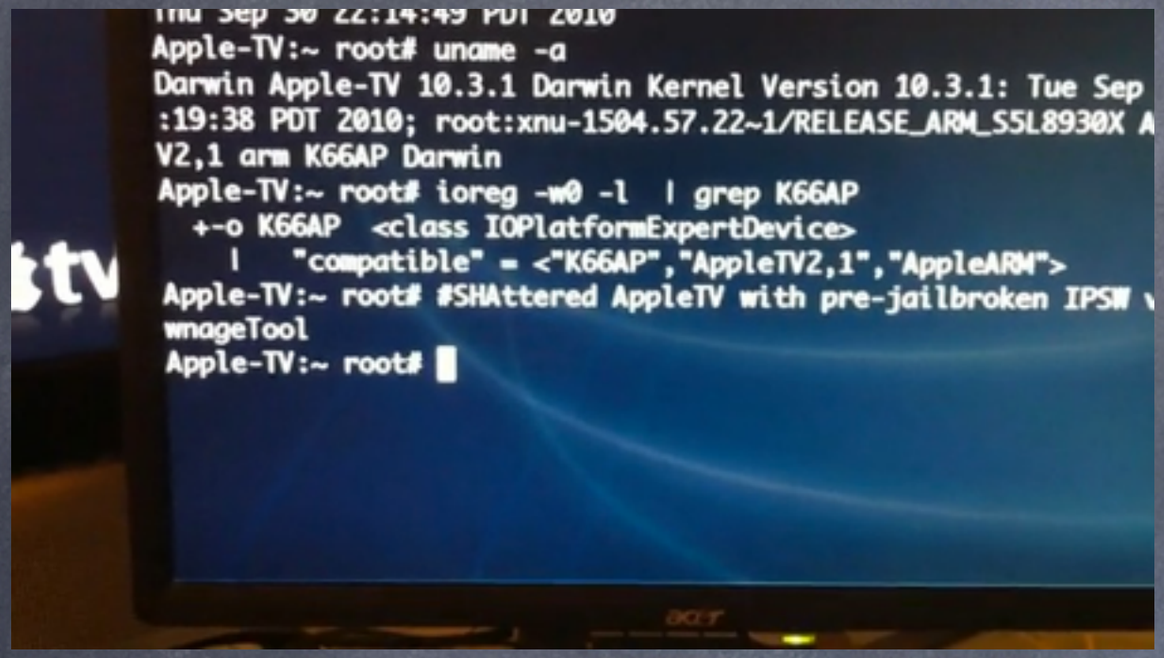
Important People (cont...)

- Brandon Holland (Couch Surfer, Skype appliance, Road Trip, BREvent/IREvent emu)
- Jim Dovey (alan quatermain) awkwardTV loader, mentor, backrow developer kit
- Thomas Cool (Take two plugin loader, Overflow, SMF framework)

AppleTV 2

- Apple A4 ARM Processor, 256 MB RAM running Custom iOS 4 version
- Shipped September 1, 2010
- October 1, 2010 MuscleNerd SHAtters AppleTV 2

SHAttered ATV2



Lowtide.app/AppleTV.app on iPod by DHowett

- September 28th 2010 Dustin Howett hacks Lowtide.app from 4.1/8M89 to run on iPod 4,1 on 4.2.1 (TUAW/Gizmodo)

- Copy of AppleTV.app/Lowtide.app, dyld_cache, multiple PrivateFrameworks, and custom bootstrap to load.

- Brandon Holland created Spin-o-rama and EventInjector to make test harness useable

Demo Of Test Harness

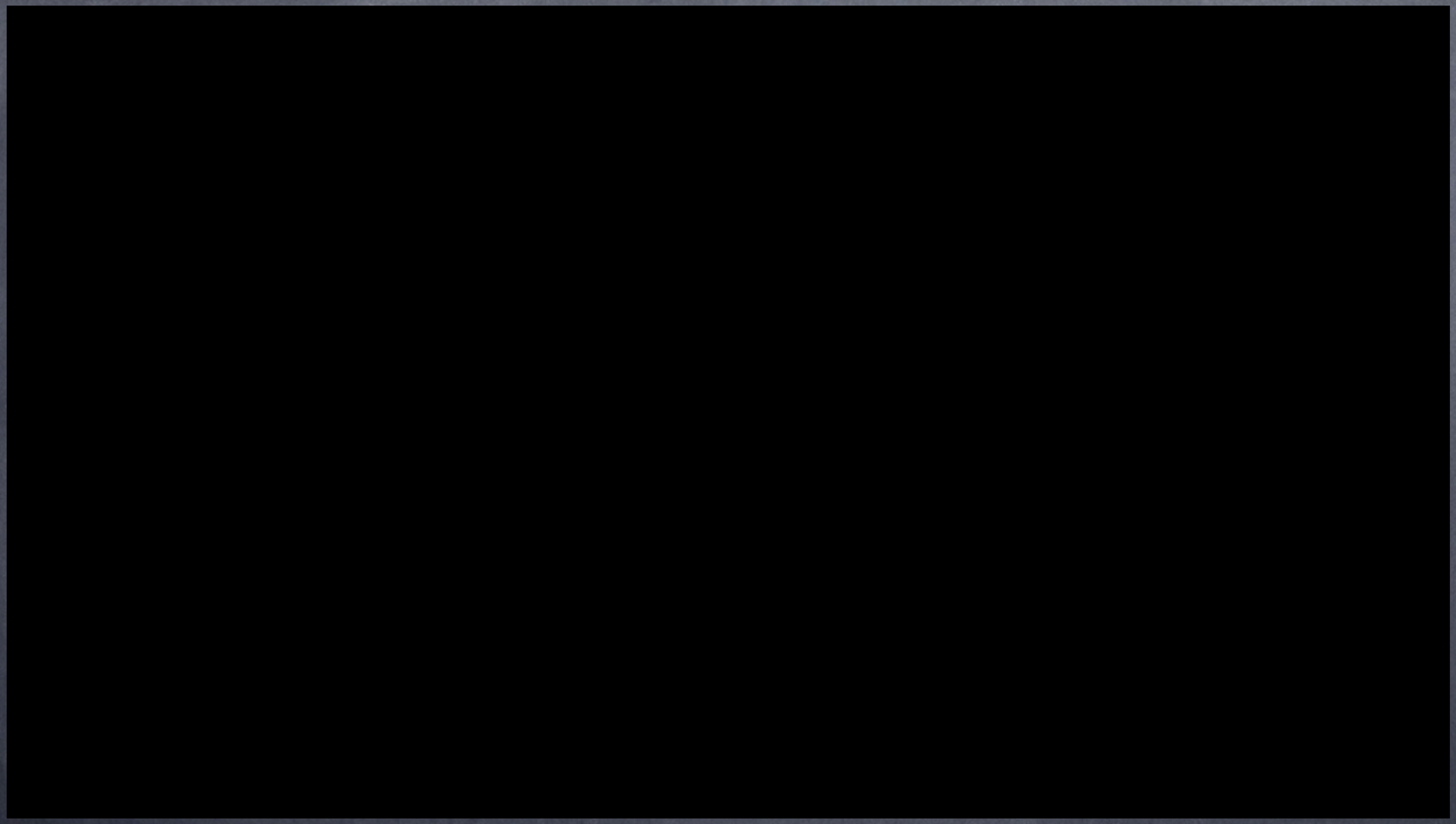
Important People

- Dustin Howett (beigelist - whitelist injector)
- Brandon Holland (spin-o-rama, event injector, exposed VNC client)
- Scott Davilla (XBMC)
- Firecore team (Couch Surfer, Media, Last.fm plugin, etc)
- Tomcool (interface tweaks, Overflow, SMF)

Jailbreaks

- October 20th, 2010: Initial Jailbreak PwnageTool 4.1 - iPhone Dev team
- February 18th, 2011: First and currently only public ramdisk based jailbreak for AppleTV in greenpois0n RC 6.1
- seas0nPass and sn0wbreeze cover most/all subsequent AppleTV 2 jailbreaks

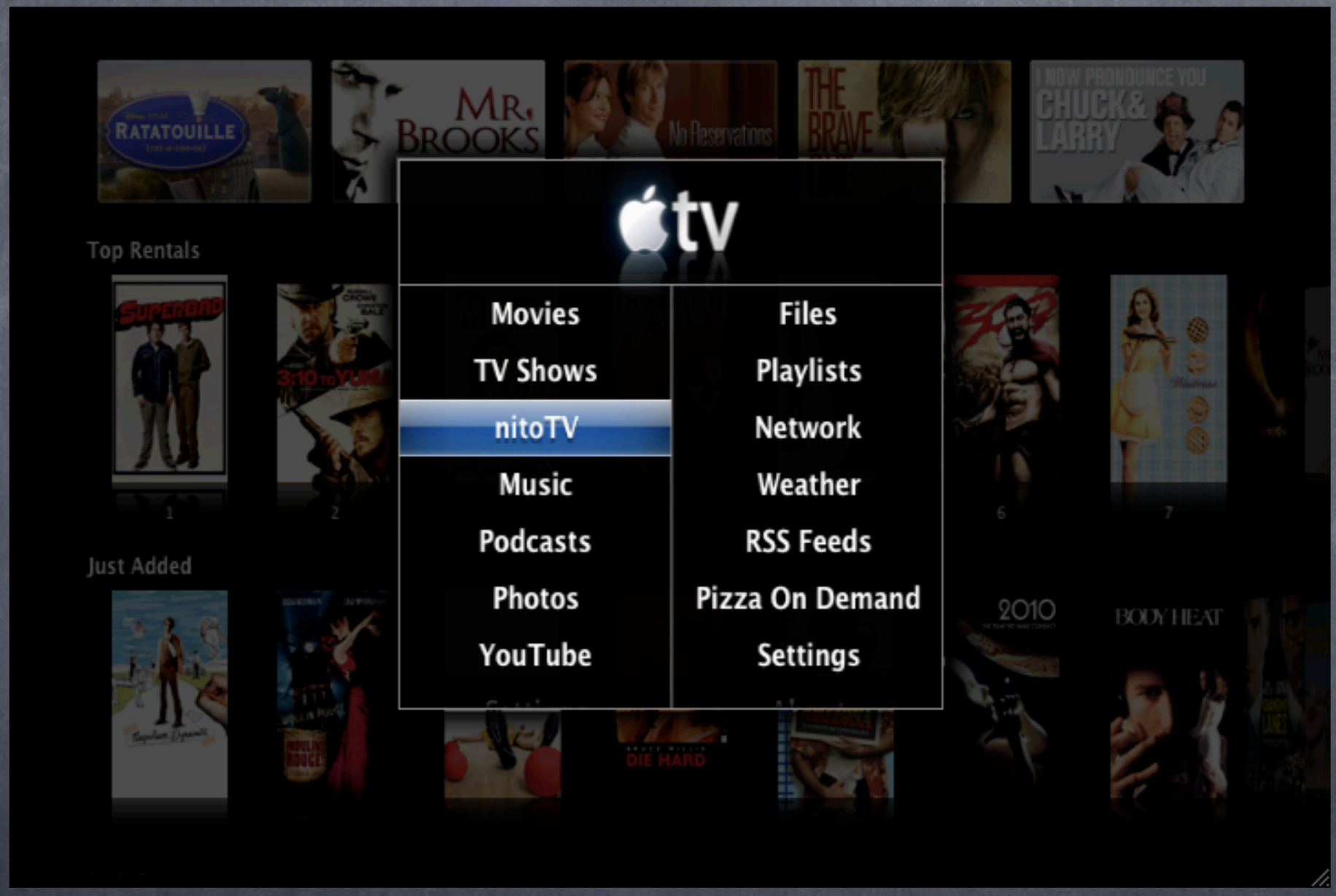
greenpois0n loader



Take One


Back Row
TV
Movies
TV Shows
YouTube
Music
nitoTV
Podcasts
Photos
Applications
Settings
Awkward TV

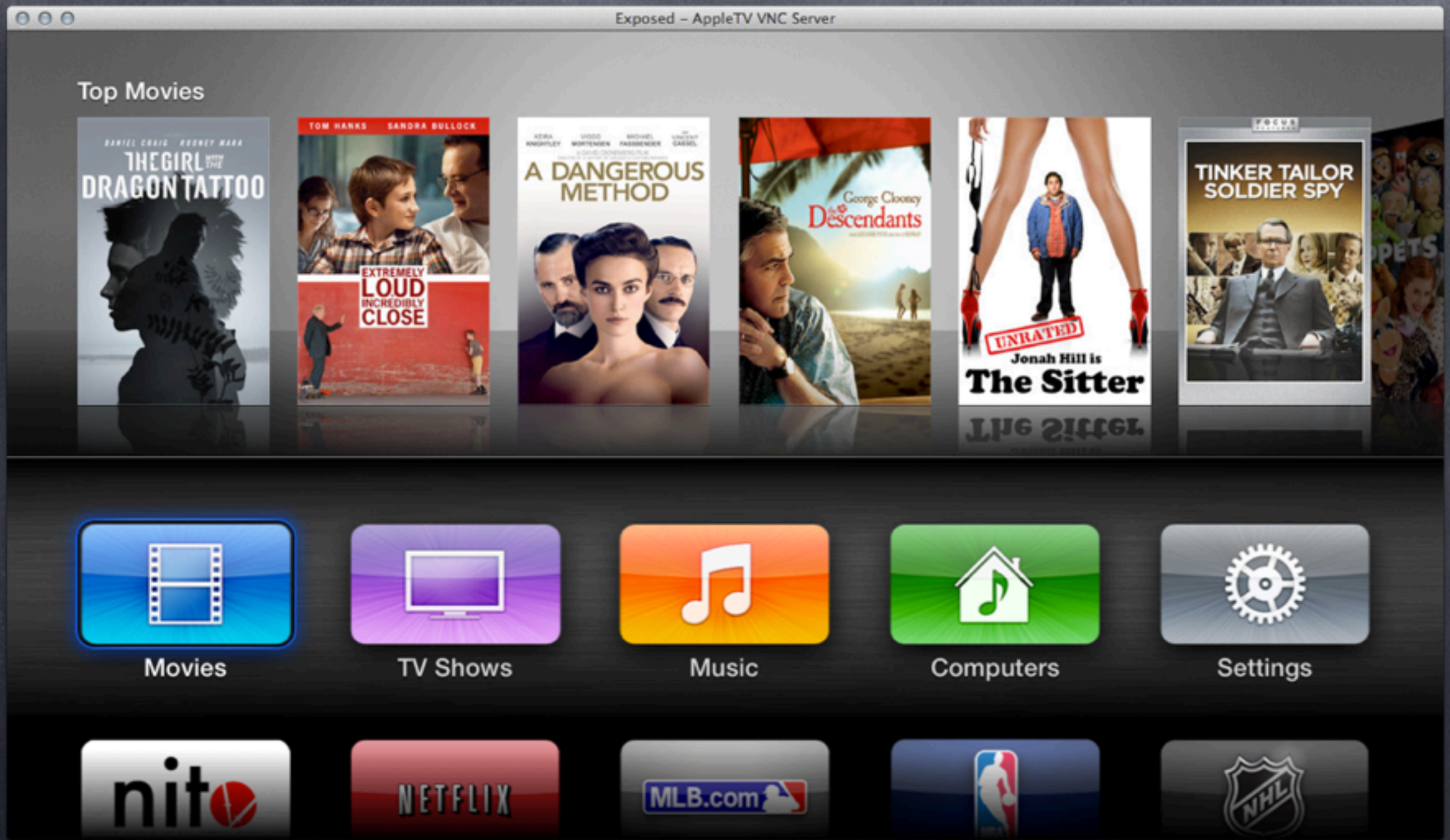
Take Two


RATATOUILLE
MR. BROOKS
No Reservations
THE BRAVE
I NOW PRONOUNCE YOU CHUCK & LARRY
Top Rentals
SUPERBAD
3:10 to YUMA
Just Added
DIE HARD
2010
BODY HEAT
tv
Movies
TV Shows
nitoTV
Music
Podcasts
Photos
YouTube
Files
Playlists
Network
Weather
RSS Feeds
Pizza On Demand
Settings

Take Three


Exposed – AppleTV VNC Server
1:15 AM
nito
Movies
TV Shows
nitoTV
Music
Internet
Computer
Install Software
Weather
RSS Feeds
Manage
Settings
About
tv

Current Interface



iOS 5.1/6.0 Notes

- AppleTV private frameworks merged into AppleTV.app
- Most variables marked private
- most plugins / tweaks require a re-write
- once released will be tethered till iPhone 5 jailbreak is done
- nitoTV still requires massive re-writing.

Example iOS 6 plugin

```
static char const * const kSMFCPDSCApKey = "SMFCPDSCAp";

static BOOL _finished = TRUE;
static int _returnCode = 0;


%subclass SMFComplexProcessDropShadowControl : SMFComplexDropShadowControl


%new - (id)ap {

    return [self associatedValueForKey:kSMFCPDSCApKey];
}

%new - (void)setAp:(id)theAp {

    [self associateValue:theAp withKey:kSMFCPDSCApKey];
}

-(id)init
{
    self=%orig;
    _returnCode=YES;
    _finished=NO;
    return self;
}
-(void)controlWasActivated
{

    %orig;
    [self performSelectorInBackground:@selector(runProcess) withObject:nil];
}
-(void)dealloc
{
        //self.ap=nil;
    %orig;
}

%new -(int)returnCode {

    return _returnCode;
}
```

AppleTV 3 Notes

- Still no Injection Vector
- On 5.x / earlier could potentially use Absinthe Exploits once injection vector found
- No official progress / ETA on anything.
- Should still probably try to stay on iOS 5.x

Q&A