



# Hacking Internet Kiosk's

Paul Craig

Defcon 16 – Las Vegas

- Who am I?
  - Paul Craig
  - Principal Security Consultant.
  - Security-Assessment.com, Auckland, New Zealand
  
  - Devoted Hacker
  - Shameless Alcoholic
  
  - Email: [paul.craig@security-assessment.com](mailto:paul.craig@security-assessment.com)
  - Web: <http://www.security-assessment.com>



- Hacking Kiosks:
  - What is an Internet Kiosk.
  - Kiosk Software Security Model.
  - Vulnerabilities in Kiosk Software.
  - Vulnerabilities in the Kiosk Security Model.
  
- Hack any Windows Kiosk in less than 120 seconds!
  
- Defcon 16 Exclusive: Tool Release.
- Live Demo's: Hacking (Two) Commercial Internet Kiosks.



- Last Year I Was Sitting in an Airport....
  - 8 hour stop-over in Hong Kong.
  - Queue of people waiting to use a hub of Internet Kiosks.
  - "Damn, those kiosks sure are popular..."
  - "I wonder if I could hack it?. Lemon party the airport.?"
  - Kiosks are popular, but rarely appear in security publications.
  - Popularity + Poor Security Visibility = Good Attack Target
- Personal Objective:
  - Find every possible method of hacking an Internet Kiosk.
  - Become the **King** of Internet Kiosk Hacking!



## What Is An Internet Kiosk

- Kiosks are everywhere
  - Airports, Train stations, Libraries, DVD Rental Stores, Corporate Building Lobbies, Convenience Stores, Post Office, Café's, Hospitals, Motels, Hotels, Universities.
  - All over the strip, even in this hotel!
  - Cheap technology has made Internet Kiosks very common.





- Initial Observations of Kiosks
  
- Hardware.
  - Kiosks often built in tough hard-shell cases.
  - Fibreglass or wooden shell.
  - Lack of physical access to the computer case.
  - Input devices hidden (Floppy/DVD/USB/FireWire)
  - Kiosk bolted to the ground (padlocked).
  
- General public are not trusted.
- Kiosk is designed to prevent theft or malicious use.



- Software.
  - Majority of Kiosks run commercial Kiosk software on Windows.
  - Linux/BSD Kiosks exist, Windows more popular.
  - 44 commercial Windows Kiosk products on the market.
  - Marketed as : "Turn that old PC into instant revenue!"
  - Buy \$59.99 Shareware -> Install -> Instant Kiosk!
- Kiosk Software Essentially Skins Windows:
  - Kiosk browser implements standard Internet Explorer libraries.
    - WINHTTP.DLL/MSINET.OCX
  - Windows drastically skinned to look like a Kiosk.
  - Still 'feels' like Windows.



- “Kiosk Software Is The Best Attack Target.”
  - Hardware hacking too obvious/intrusive for public locations.
- “I Need to Walk up to Any Internet Kiosk and Pop Shell, Quickly.”
  - Explorer.exe, cmd.exe, command.com.
  - Time limited, 5 minutes or less.
- 16 Months of Kiosk Software Penetration Testing.
  - Virtualized the top ten most popular Windows Kiosk products.
  - Detailed and compared the security model of each Kiosk product.
  - Researched new methods of compromising internet Kiosks.
  - Developed **Kiosk Attack Methodology**.
  - Startling results: 100% success rate!

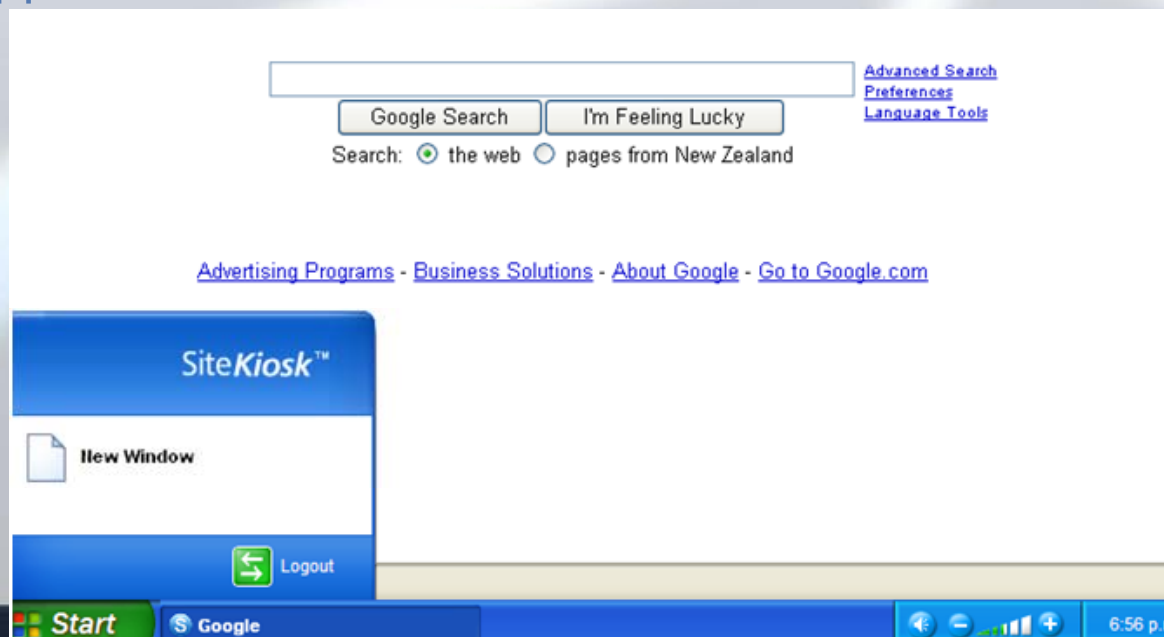


# Kiosk Security Model



- Kiosk Software Implement Security by Two Approaches.
  - Reduce Available Host Functionality.
    - Disallow native OS functionality that can be used maliciously.
    - “Command Prompt has been Disabled”
    - Implemented through native ACL's.
  - Jail Users Into a 'Secure Kiosk Browser' Application.
    - Users are stuck inside a Kiosk browser.
    - Kiosk browser ran in full screen, no ability to close, minimize.
    - Start Bar/Tray Menu removed or hidden.
    - Only thing you can do is browse the web.

- Example #1: Site Kiosk.
  - Looks similar to Windows.
  - Custom Tray Menu/Task Bar.
    - Only one option, 'New Window'
    - Real Windows 'Start' bar is hidden from view.
  - Trapped inside the Kiosk browser.



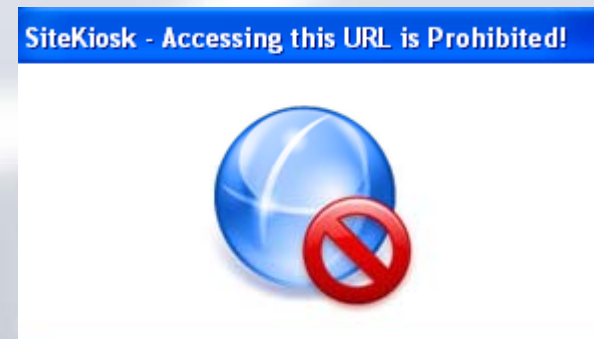


- Example #2: NetStop Kiosk
  - Custom task bar, no buttons at all.
  - Kiosk application ran as a full screen desktop.
  - No ability to close the browser.
  - Only permits internet browsing.
  - Used here at the Riviera hotel





- Kiosk Browsers Proactively Monitor Your Activity.
  - Kiosks contain numerous blacklists of prohibited activity.
  - Try to do something sneaky, the Kiosk will stop you.
- Try to Browse C:\ with the Kiosk browser:
- Monitor in-focus Modal Dialogs.
  - Block dialogs by Window Title or Window Class.
  - "Save File As", "Open With", "Confirm File Delete", "Print".
  - Kiosk sends WM\_CLOSE Message to the blacklisted dialogs.





- API Hooking.
  - Hook native OS API calls which can be used maliciously.
  - KillProcess(), GetCommandLineW(), AllocConsole()
  - “Unauthorized Functionality Detected, Process Killed”.
- Kiosk Browser ran in ‘High Security Zone’
  - File downloads disabled.
  - Browser scripting, pop-ups, ActiveX, all disabled.
- Watchdog Timer.
  - Every X minutes the Kiosk will enumerate all active processes.
  - Terminate any unauthorized processes.



- Custom Keyboard Driver.
  - Disable Windows shortcut key combinations.

|                           |
|---------------------------|
| CTRL-SHIFT-ESC (Task Mgr) |
|---------------------------|

|                       |
|-----------------------|
| ALT-TAB (Switch Task) |
|-----------------------|

|                            |
|----------------------------|
| CTRL-ALT-DELETE (Task Mgr) |
|----------------------------|

|                       |
|-----------------------|
| CTRL-ESC (Start Menu) |
|-----------------------|

|                            |
|----------------------------|
| Alt-F4 (Close Application) |
|----------------------------|

- Modifier keys unmapped.
  - CTRL, Tab, ALT, 'Start', Function, F1-F12.
  - Custom Keyboard with missing modifier keys!
- Custom Mouse.
  - No right click button.
- All Methods of reducing functionality!





# Hacking Kiosk Software



- Kiosk Security Model is Based on Reducing Functionality.
  - Limit functionality which can be used to escape the Kiosk browser.
- Exploiting A Kiosk Requires **Invoking Functionality**.
  - Cause applications/functionality to spawn, popup on screen.
  - Use the invoked functionality to escape the Kiosk jail.
  - Spawn a command prompt, get back to Windows.
- Kiosks Implement Blacklists.
  - Blacklists (by nature) are never 100%.
  - We only need one method of escaping the software jail.
- Kiosks are very hack-able.



- What Are The Available Kiosk Input Vectors?
  - #1 - **Physical Input:**
    - Interacting with the Kiosk GUI.
    - Using the keyboard or mouse.
    - Clicking on buttons, graphics, menu's.
    - Type values into text input fields.
  - #2 - **Remote Input:**
    - Browsing a website from a Kiosk terminal.



- What Do We Need To Do?
  - **#1 - Escape The Kiosk Graphical Jail.**
    - Minimize or close the Kiosk browser application.
    - Pop a command shell.
    - Kill the Kiosk browser process (`taskkill /IM KioskBrowser.exe`)
    - Re-Enable the hidden (real) Windows Start bar.
    - Get "Back to Windows."
  - **#2 - Download Additional Binaries to The Kiosk.**
    - Port scanner, Metasploit, rootkit, trojan, keylogger.



- You Find a Kiosk in Your Local Mall.
  - "\$1 for 1 hour of internet usage"
  - Insert a dollar.
- You Find You are Trapped Inside a Kiosk Browser.
  - Right mouse button has been disabled.
  - Custom keyboard with certain keys removed.
  - Feels like a Windows OS , but has a custom design/layout.
    - 'Start' bar is labelled 'Super-Kiosk Start'.
  - Only one visible button to 'Start Browsing'
  - Start Browsing...

- Browse The Local File System Using The Kiosk Browser.
  - Local Windows users are capable of browsing the file-system.
  - Kiosk software must explicitly block local browsing attempts.



- Windows is designed for idiots.
  - Caters for mistypes/fat-fingers.
  - C:\windows\ maybe blocked.

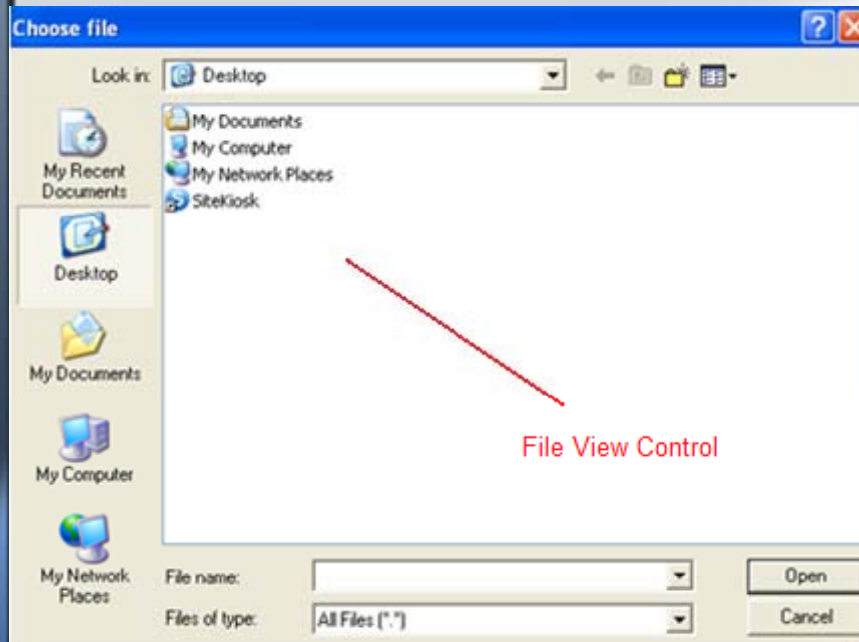
|                   |                    |                   |                  |
|-------------------|--------------------|-------------------|------------------|
| File:/C:/windows  | File:/C:\windows\  | File:/C:\windows/ | File:/C:/windows |
| File://C:/windows | File://C:\windows/ | file://C:\windows | C:/windows       |
| C:\windows\       | C:\windows         | C:/windows/       | C:/windows\      |
| %WINDIR%          | %TMP%              | %TEMP%            | %SYSTEMDRIVE%    |
| %SYSTEMROOT%      | %APPDATA%          | %HOMEDRIVE%       | %HOMESHARE%      |

- Blacklists start failing about now.

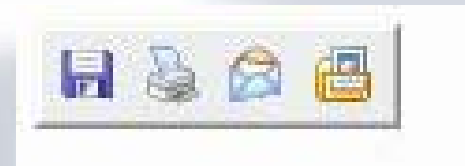


- Using Common Dialogs To Hack Kiosks.
  - Windows contains 'Common Dialogs' libraries.
    - Saving a file, opening a file, selecting font, choosing a colour.
    - COMDLG32.DLL (Common Windows Dialogs Library).
    - COMDLG32.DLL Implements Common Windows Controls.
      - From COMCTL32.DLL.
  
- File/Open, File/Save Dialog's Contain 'File View' Controls.
  - File view control provides full Explorer functionality.
  - Same control that Windows Explorer uses.
  - File-Open Dialog = Explorer
  - Can be used to launch processes.

- Systematically Click Every Button, Graphic, Icon In The Kiosk
  - Can we invoke a File - Open Dialog? "Attach File"
  - Browse the file system
  - Right Click cmd.exe: Open / Run As
  - Spawn cmd.exe



- Internet Explorer 'Image Toolbar'.
  - Toolbar hovers top-left of a large image when clicked.
  - Each icon of this toolbar can invoke a Common Dialog.
    - File/Save.
    - File/Print.
    - File/Mailto.
    - Open "My Pictures" in Explorer.
- Toolbar is present if the Kiosk uses Internet Explorer libraries.
- Click a large image on screen
  - Spawn a Common Dialog, spawn Explorer.



- Using the Keyboard.
  - Keyboard shortcuts can be used to access the host OS.
  - Check if a custom keyboard driver present?
    - Are modifier keys enabled?

- Keyboard Shortcuts Which Produce Common Dialogs.

|                                    |
|------------------------------------|
| CTRL-B, CTRL-I (Favourites)        |
| CTRL-H (History)                   |
| CTRL-L, CTL-O – (File/Open Dialog) |
| CTRL-P – (Print Dialog)            |
| CTRL-S – (Save As)                 |

- Kiosk Specific 'Administrative' shortcuts.
  - All Kiosk products contain a hidden Administrative menu.
  - Mash the keyboard, CTRL-ALT-F8? CTRL-ESC-F9?



- Browser Security Zones
  - Browser security model incorporates multiple security zones:

**Restricted Sites**

**Internet Zone**

**Intranet Zone**

**Trusted Sites**

- Each security zone adheres to a unique security policy.
  - Internet zone has less ability to interact with the Kiosk.
  - Locked Down by the Kiosk browser.
  - Trusted Sites, Intranet Zone have more access.

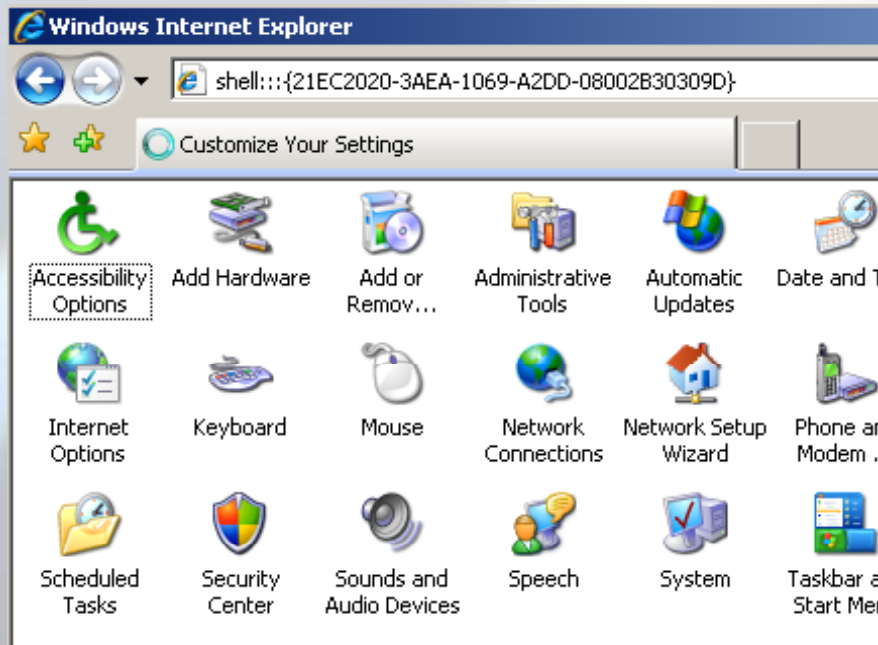


- Local Users Can Access All Available Security Zones.
  - URL's must be typed into the URL entry bar.
- Security Zone Escalation: about: pluggable-protocol handler.
  - About handler belongs to the 'Trusted Sites' security zone.
  - Suffers from a Cross Site Scripting vulnerability.
  - Local user can render arbitrary content within a trusted zone.
  - Spawn a File Open Common Dialog from a trusted security zone.  
**about:<input%20type=file>**  
**about:<a%20href=C:\windows\>Click-Here</a>**
  - Internet Zone may be locked down, Trusted Sites never are.



- Shell Protocol Handler.
  - Shell handler provides access to Windows web folders.
  - Type Into the URI Bar:
    - Shell:Profile
    - Shell:ProgramFiles
    - Shell:System
    - Shell:ControlPanelFolder
    - Shell:Windows
  - Each URL will spawn explorer.exe and browse the webfolder.
  - Is the shell: handler blocked by the Kiosk?

- How About This:
  - `shell:::{21EC2020-3AEA-1069-A2DD-08002B30309D}`
  - Web folder invoking the Windows Control Panel by CLASSID.
  - Works from WININET.DLL/MSINET.OCX
  - Bypass ACL's that may exist on control.exe (Control Panel)





- The Downside to Physical Input Vectors.
  - Kiosk software is designed to not trust the guy on the keyboard.
  - **Kiosk User = Most Obvious Security Threat.**
  - My research concluded that physical inputs are not very successful.
    - 40-50% chance of popping shell.
    - Most of these techniques already published, unoriginal.
  
- Then I Made a Subtle Discovery...
  - Remote websites **not** factored highly into the Kiosk security model.
  - Websites are often trusted **MORE** than a local Kiosk user!
  - Kiosks rely on the default browser security model!
  - Not designed for Kiosks.



- Available Remote Input Vectors:
  - Content visited from a Kiosk terminal.
  - Hosted on a remote website.

|                             |                    |           |                      |
|-----------------------------|--------------------|-----------|----------------------|
| Browser Scripting Languages | Java Applets       | ActiveX   | ClickOnce (.NET)     |
| Protocol Handlers           | File Type Handlers | Flash     | Director             |
| Acrobat                     | Real Media         | QuickTime | Windows Media Player |

- More input vectors than physical attacks!



- “I Need a Kiosk Hacking Website.”
  - An online tool you can visit from an Internet Kiosk.
  - Provides content to escape a Kiosk jail.
- iKAT – Interactive Kiosk Attack Tool – Official Release
  - First of its kind! New method of hacking Internet Kiosks!
  - Fast! iKAT can pop shell in less than 120 seconds .
  - 95-100% success rate!!



- Officially Launched @ Defcon 16: <http://ikat.hacked.net>



- What Can iKAT Do?
- Kiosk Reconnaissance : Detect Installed Applications
  - JavaScript & res:// (resource) protocol handler.
  - Extract bitmap resources from PE executables.
  - Verify bitmap presence and detect installed applications.
  - Detects the most common commercial Kiosk platforms.
  - Enumerates locally installed applications.

```
var disk;  
disk = 'C:\\';  
var test = new Image();  
test.src = 'res://C:\\' + fileurl;  
if (test.height != 30)  
{  
return true;  
}
```

```
Detected Kiosk Platform:  
NetStop Pro Kiosk          C:\Program Files\NetStopPro\  
  
Detected Applications:  
Windows Media Player 11   C:\Program Files\Windows Media  
Microsoft NetMeeting      C:\Program Files\Netmeeting\  
Microsoft .NET Framework v1.0 C:\Windows\Microsoft.NET\Fram  
Microsoft .NET Framework v2.0 C:\Windows\Microsoft.NET\Fram  
MSN Messenger              C:\Program Files\Messenger\  
Microsoft Movie Maker     C:\Program Files\Movie Maker\  

```



- Display Local Browser Variables.
  - Determine underlying Kiosk browser technology.
  - MSINET.OCX, WINHTTP.DLL display Internet Explorer appVersion
  - Detect the presence of .NET CLR.

```
Local Browser Variables

Navigator.appName
Microsoft Internet Explorer

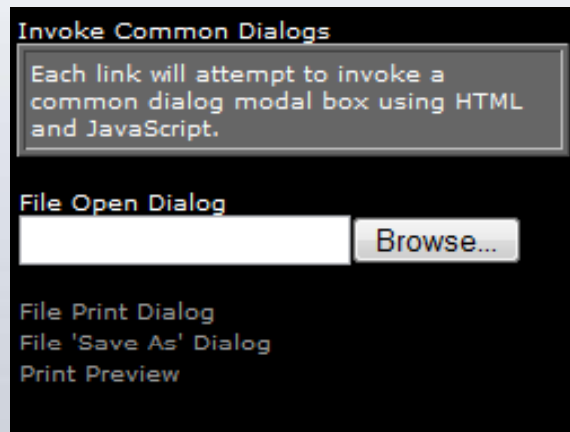
Navigator.appVersion
4.0 (compatible; MSIE 7.0; Windows NT
5.1; .NET CLR 2.0.50727)

Navigator Platform
Win32

Navigator Useragent
Mozilla/4.0 (compatible; MSIE 7.0;
Windows NT 5.1; .NET CLR 2.0.50727)
```

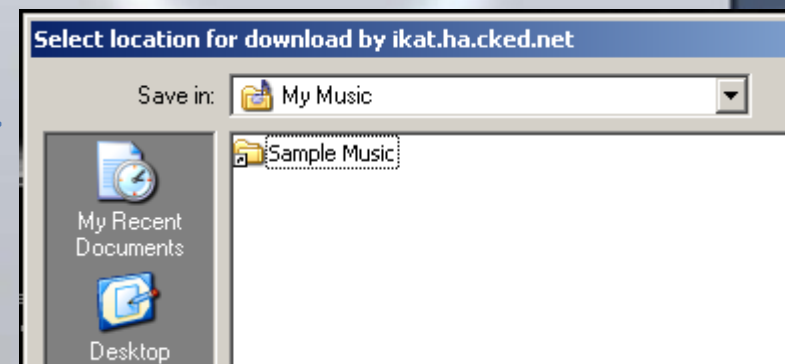
- Display Remote Server Variables
  - Discover remote IP address of the Kiosk terminal.

- All Common Browser Dialogs In One Place



- File Open, Save As, Print, Print Preview:
- Click down the list and determine what dialogs are blocked.
  - Use the File View control within the dialogs.

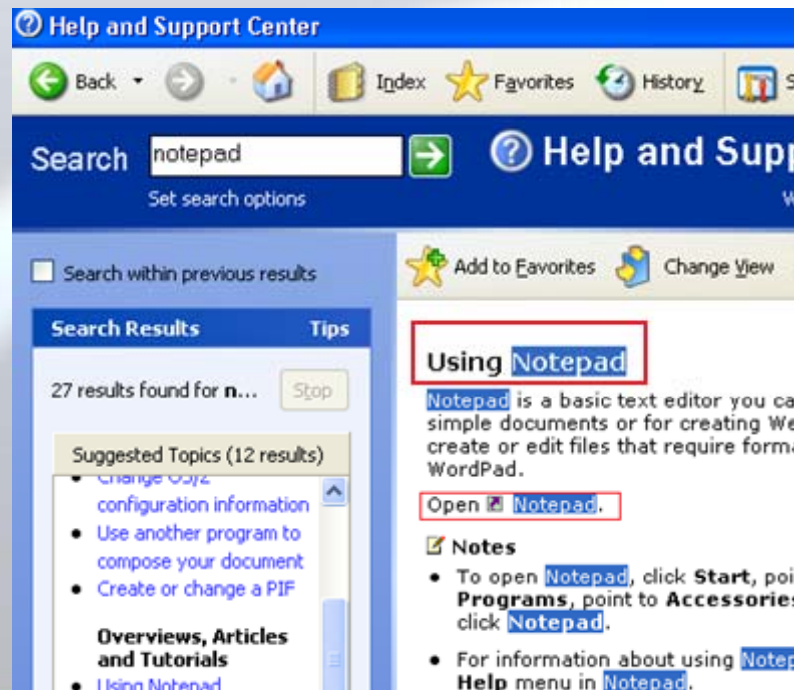
- Use Flash To Invoke Common Dialogs.
  - Adobe Flash is the most widely used browser plug-in.
  - ActionScript 3 can invoke three unique File Open dialogs.
    - 'Select File For Upload'
    - 'Select File(s) For Upload'
    - 'Select location for Download by ikat.ha.cked.net'
- Flash Common Dialogs have Unique Dialog Titles
  - Not standard "Choose File"
  - Bypass dialog Window title blacklists.
  - Still contain the File View control.





- Spawning Applications On The Kiosk.
  - Can we cause an applications/processes to launch on the Kiosk.
  - Perhaps the spawned application contains a common dialog?
  - Application provides additional access to the Kiosk.
- iKAT Invokes Default Windows URI Handlers.
  - Callto://, Gopher://, HCP://, Telnet://, TN3270://, Rlogin://, LDAP://, News://, Mailto://
    - **<A HREF=Mailto://aaa> Mailto </a>**
    - Outlook express spawns.
- 3<sup>rd</sup> party URI Handlers
  - MMS://, SKYPE://, SIP://, Play://, Steam://, Quicktime://

- Example: HCP://: Help And Support Center
  - `<a href=HCP://dummy> Click-me </a>`
  - Search HCP for what you want to launch.
  - “Using Command Prompt” provides link to spawn cmd.exe
  - Left Click Only! (No right click button required)









- iKAT Provides Links to over 100 URI handlers.
  - Click, click, click down the list.
  - Determine which handlers are covered by the Kiosk blacklist.
  - Use invoked handler application to escape the Kiosk.

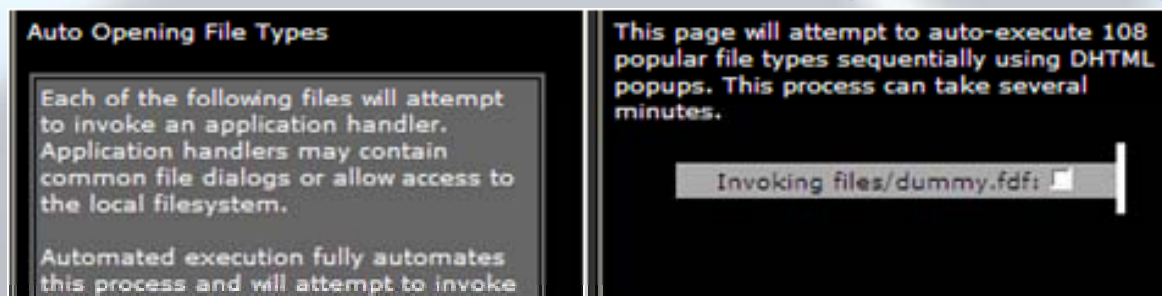
- iKAT Contains Local Zone Handlers
  - about:, res:, shell:
  - Lists of URL's to type in.
  - Remembering ClassID's is hard.

```
aolautofix://      imesync://
acrobat://        icquser://
adobebridge://    ircs://
bittorrent://     itms://
camfront://       itmss://
daap://           itpc://
ed2k://           joost://
fdaction://       mapi:// (outlook)
feed://           Mirc://
feeds:// (outlook2k7) MSNIM:// (Pidgin)
FireFox.Url://    MYIM:// (Pidgin)
FireFoxURL://     MMS:// (Media Player)
gtalk://          MMST:// (Media Player)
groove:// (outlook2k7) MSBD:// (Media Player)
gizmoproject://   MMSU:// (Media Player)
gnet://           M4MacDrive://
gnutella://       magnet://
gsarcade://       mediajukebox://
IE.FTP://         Morpheus://
IE.HTTP://        Mozilla://
IE.HTTPS://       mp2p://
irc://            mpodcast://
ICY://            News://
```

- Invoke Applications Using File Type Handlers.
  - Click on test.myfile, Windows will spawn the 'myfile' handler.
  - Internet Explorer supports prompt-less handler execution.
    - Example: Click test.wmv, Windows Media Player Spawns.
    - No Prompt "Are you sure you want to...".

|   |                |                                   |
|---|----------------|-----------------------------------|
|  (Default)               | REG_SZ         | Windows Media Player Skin Package |
|  EditFlags               | REG_BINARY     | 00 00 01 00                       |
|  FriendlyTypeName        | REG_EXPAND_... | @%SystemRoot%\system32\unregm...  |
|  PreferExecuteOnMismatch | REG_DWORD      | 0x00000001 (1)                    |

- Kiosk blacklists monitor all in focus dialogs for warning prompts.



- iKAT uses DHTML/JavaScript to invoke 108 unique file handlers.

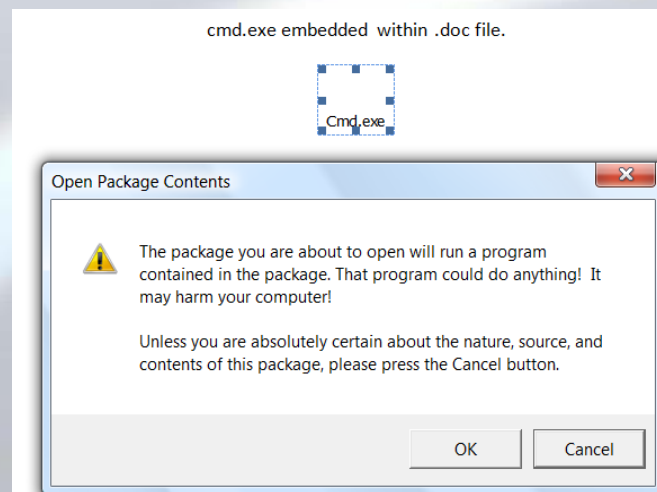


- iKAT & Windows Media Files.
  - WMPPlayer will silently launch for multiple file types.
  - Windows Media Playlist Files (.ASX)
  - Supports 'Web Enhanced Content'.
  - Turns Windows Media Player into a web browser!
  - Provides a browser without the typical Kiosk security controls.

```
<ASX VERSION="3.0">
<PARAM name="HTMLView" value="http://ikat.ha.cked.net/">
<ENTRY>
  <REF href="http://ha.cked.net/front.jpg"/>
</ENTRY>
</ASX>
```



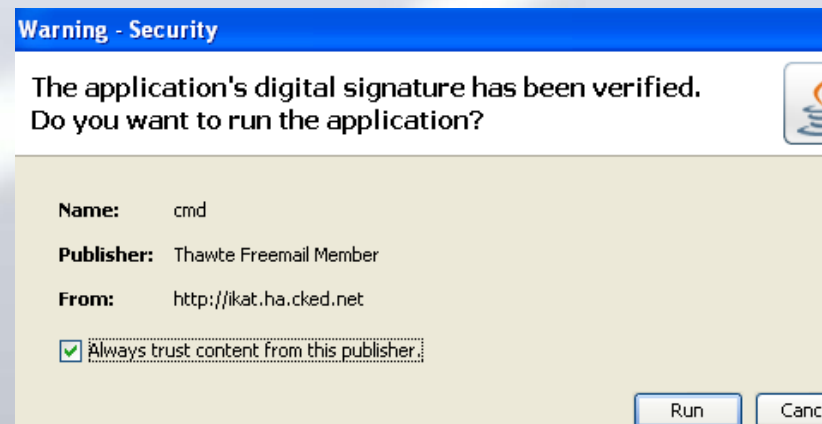
- iKAT & Office Documents.
  - If an Office file viewer is installed on the Kiosk, we win.
  - Embed a copy of cmd.exe within an office document.
  - Supported by .DOC, .DOCX, .XLS, .XLSB, .XLSM, XLSX
- Using Word to Execute Our Embedded CMD.EXE.
  - 'Open Package Contents' dialog not detected by any Kiosk.



- iKAT will spawn the most 'useful' file possible.

- iKAT & Java Applets:
  - Signed Java applets can execute local processes.
  - Detect if JRE is installed (iKAT Kiosk Reconnaissance Trick).
  - Does the Kiosk detect the Java security warning prompt?

- "Warning – Security"



- iKAT Contains Signed Kiosk Specific Java Applets.
  - Signed applets to spawn command shells
  - Includes Jython by GNUCITIZEN



- Install a Malicious ActiveX
  - Safe for scripting ActiveX's can be used to compromise a Kiosk.
  - Unsafe method: `object.execute('cmd.exe');`
  - Can we install a malicious ActiveX on the Kiosk?
- iKAT ActiveX
  - Safe-for-scripting ActiveX which executes arbitrary executables.
  - Installing an ActiveX requires administrative authority.
- ActiveX is changing:
  - Internet Explorer 8 does not require admin rights for ActiveX.

- iKAT & ClickOnce Applications

- ClickOnce is .NET 2.0+ technology (.NET CLR 2+ required)
- 'Online Application Deployment' .application file handler.
- Unsigned ClickOnce applications execute with full trust!
- Admin privileges are not required!

- Users are warned:



- 100% of tested Kiosks fail to detect this warning message
- Modern Kiosks now developed in .NET (CLR is present!)



- “What are the most useful ClickOnce applications to have?”
- **iKAT - Embedded Web Browser.**
  - HTTP browser with reduced security settings.
- **iKAT - Application Executor.**
  - Spawn's any arbitrary executables on a Kiosk.
- **iKAT - Access Token Pincher**
  - Access token hijacking is very hip subject, why not.
  - Does the Kiosk user have the SeImpersonate privilege?
  - Impersonate available (privileged) tokens.
  - Spawn cmd.exe under the context of the privileged token.
  - Pop a system shell!



- Who Here Has Ever Crashed a Web Browser?
  - What about crashing a Kiosk: 'Emo-Kiosking'
  - Create an unhandled exception in a Kiosk browser.
  - Kiosk browser crashes, We get the desktop, We Win!
  - Rare situation: Application crash = highly critical vulnerability.
- iKAT Contains Common Browser Crash Techniques.
  - Published exploits which results in a crash.
  - Fastest, easiest method of escaping a Kiosk.
  - Fairly reliable, 40% of tested Kiosks crash.

### Crash a Kiosk

Why bother exploiting a Kiosk when crashing it will give you the desktop? Create an unhandled exception and you win..

Otherwise known as 'Kiosk Self Mutilation' or Emo-Kiosking

### Previously Published Flaws

Input Type=Crash  
Java Document.Write Loop  
CSS Position  
CSS Memory Corruption  
Body onLoad="window()"  
MHTML onClick  
HTML Orderd List  
JavaScript Memory Exhaustion  
Res:// Integer Overflow  
Flash 8 IE7 Stack Overflow  
AutoMagic Flash Crash

- Crashing Browser Plug-ins.
  - Flash is the most common browser plug-in available.
  - “Can I create a .SWF file that reliably crashes a browser?”
  - Turns out yes, yes, you can!
    - File format fuzzing of .SWF found multiple crash situations.
    - Immediately un-exploitable, reliably crash any browser.
    - Created ‘iKAT Auto Magic Flash Crasher’.
- Is the Flash plug-in installed on the Kiosk?
  - iKAT can crash it, guaranteed.
  - Does the Kiosk detect the unhandled exception and re-spawn?
  - Or just present the desktop?

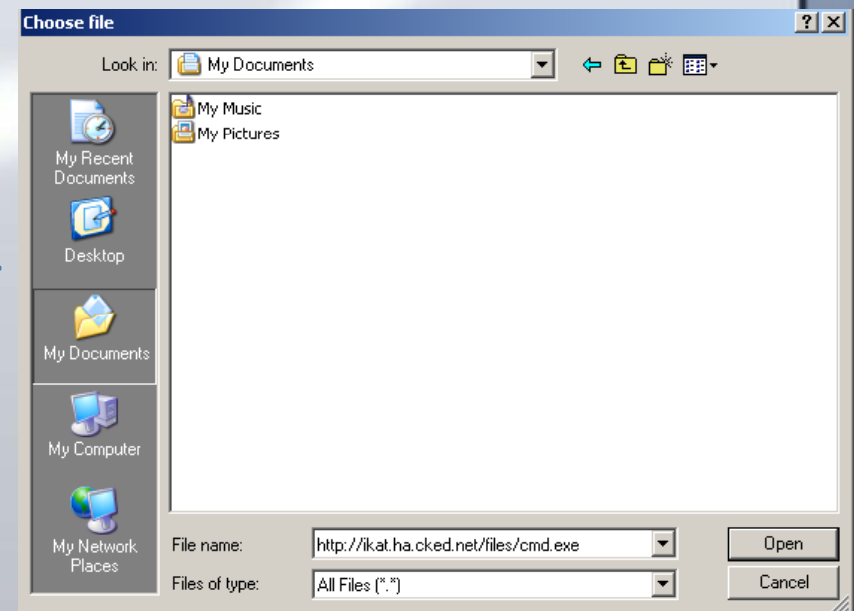


- Lets Assume Something Worked.
  - You have access to the Kiosk File system.
  - Command shell spawned, Common Dialog, Java installed, etc
  
- What Now?
  - Download additional tools/binaries.
  
- How Do You Download Files In a Tool-less Environment.
  - Kiosk terminal will not have a copy of wget.exe present.
  - Internet Explorer is likely uninstalled or disabled.
  - File downloads disabled.



- Old School: Downloading Files In Windows:
- Using Common Dialogs
  - 'Attach' a remote file from a File-Open dialog.
  - FPSE/WebDAV to save the file locally, and attach it.

- Works From Any File->Open Dialog.
  - File saved in a writeable location.
  - Temporary internet files.
  - Downloads any file type/size.

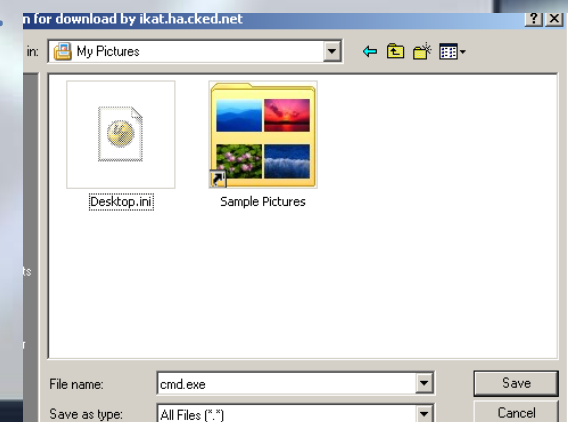


```
Directory of C:\Documents and Settings\kiosk-user\Local Settings\Temporary Int
ernet Files\Content.IE5\PMN68AXH
06/24/2008 02:39 PM          388,608 cmd[1].exe
06/24/2008 02:32 PM           1,450 ikat.ha.cked[1].htm
                2 File(s)          390,058 bytes
0 Dir(s)          5,164,800 bytes free
```

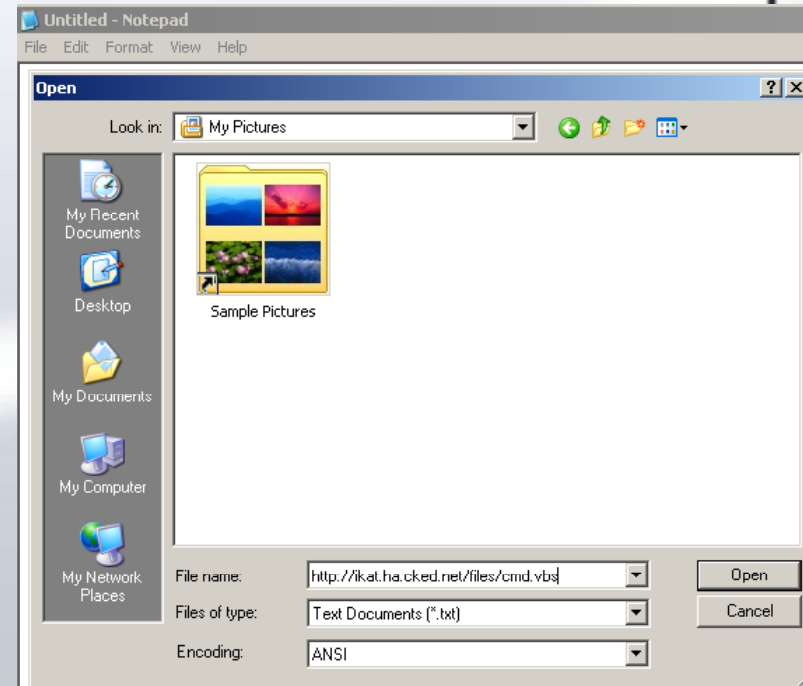
- Use Flash To Download Files.
  - Kiosk's disable File Downloads within the browser security policy.
  - IE: Tools -> Internet Options -> Custom Level



- Flash can be used to circumvent the browser policy.
  - Download method of FileReference() object.
- Flash does not validate browser security policy.
- Very high success rate against Kiosks.
- Because its an 0day trick.



- Notepad Can Download Files.
- File-> Open
  - <http://test.com/trojan.txt>
  - File downloaded.
  - Cannot save 8bit data.
  - Content must be 7bit safe.
- File-> Save
  - Upload content to a remote site.
  - FPSE/WebDav
  - <http://www.ok.com/blah.txt>
  - Quickly upload files from a Kiosk.



- #1 Problem: Kiosk Hacking is a Tool less Environment
  - “iKAT needs to provide tools for Kiosk hacking”.

- Assorted Kiosk Hacking Tools

```

Command Shells.
cmd.exe           [.exe] [.zip] [Flash]
command.com       [.com] [.zip] [Flash]

Network Tools.
Netcat            [.exe] [.zip] [Flash]
GNU WGet          [.exe] [.zip] [Flash]
Nmap              [.exe] [.zip] [Flash]

Exploitation Aids.
Enable Hidden    [.exe] [.zip] [Flash]
StartBar
Application Executor [.exe] [.zip] [Flash]
Command Shell    [.exe] [.zip] [Flash]
Detour
Group Policy Bypass [.zip] [Flash]
Hacked Kiosk Popup [.exe] [.zip] [Flash]
  
```

- All tools available in .exe, .zip, Flash security policy bypass.
  - Also Available as 7bit Safe VBScript (.VBS/.VBE)!
  - Download and save the tool using notepad.



- Command Shell Detours:
  - How many ways to spawn a command shell on Windows?

|                           |  |                              |                         |
|---------------------------|--|------------------------------|-------------------------|
| cmd.exe                   | command.com  | win.com cmd.exe              | win.com command.com     |
| Loadfix.com start.exe     | sc create testsvc binpath="cmd /K start" type= own<br>type= interact | loadfix.com cmd.exe          | loadfix.com command.com |
| start loadfix.com cmd.exe | start loadfix.com<br>command.com                                     | start loadfix.com<br>cmd.exe | %COMSPEC%               |

- Win.com? Loadfix.com? Start? Combinations of both?
- ACL's on the Kiosk may block cmd.exe
  - What about command.com, win.com?
- 'CMD Detours' tool attempts 17 methods of invoking a shell.
- Flawless at bypassing Kiosk ACL's.



- Using iKAT
  - iKAT is a tool designed to aid penetration testing/Kiosk hacking.
  - Use it to configure your own Kiosk securely!
    - Test the strength of your own blacklists
    - Blacklist Window dialogs, increase your level of security.
- Feedback Welcomed!
  - Submit a feature request, report a bug, help me improve iKAT.
- iKAT is going Open Source!
  - **iKAT Portable** being released soon
  - Downloadable version you can host locally, memory stick.
  - Useful for security consultants without internet access.

# Hacking Kiosks : The Demo's

- Two virtualized (commercial) Kiosk products.
- Recommended Kiosk application configuration.
- Default Windows XP install.
  
- Using iKAT To Pop a Command shell
  - As Fast As Possible!



Happy Hacking.

Questions?

Email me:

[paul.craig@security-assessment.com](mailto:paul.craig@security-assessment.com)