SCREEN SCRAPER TRICKS DIFFICULT CASES
Define what Constitutes a “DIFFICULT CASE”
Demo some "SCREEN SCRAPER TRICKS"

TODAY'S AGENDA
Look at ideas for LARGE-SCALE DEPLOYMENT.

TODAY'S AGENDA
Today's Agenda

Share a Heartwarming Moment
Share a HEARTWARMING MOMENT

Featuring CAPTCHAs!

TODAY'S AGENDA
Goals of this Talk

Gain an understanding of some unusual (useful) web scraping techniques

Your not going to walk away form here with ready-made solutions

The goal is to expose you to some new ideas that you can apply to your specific situation
Goals of this Talk

Gain an understanding of some unusual (useful) web scraping techniques

Your not going to walk away form here with ready-made solutions

The goal is to expose you to some new ideas that you can apply to your specific situation
Goals of this Talk

Gain an understanding of some unusual (useful) web scraping techniques

Your not going to walk away form here with ready-made solutions

The goal is to expose you to some new ideas that you can apply to your specific situation
Technologies & Tools Discussed

● For the purposes of this discussion, the solutions have to meet three criteria:
Technologies & Tools Discussed

- For the purposes of this discussion, the solutions have to meet three criteria:
  #1. Completely customizable (hackable)
Technologies & Tools Discussed

• For the purposes of this discussion, the solutions have to meet three criteria:

#1. Completely customizable (hackable)

#2. Free (or Open Source)
Technologies & Tools Discussed

• For the purposes of this discussion, the solutions have to meet three criteria:
  #1. Completely customizable (hackable)
  #2. Free (or Open Source)
  #3. Platform independent
Michael Schrenk

BIO:

• Minneapolis-based bot writer, consultant & author
Michael Schrenk

BIO:

- Minneapolis-based bot writer, consultant & author
- (Soon to be) Las Vegas-based
Michael Schrenk

BIO:

• Minneapolis-based bot writer, consultant & author
• (Soon to be) Las Vegas-based
• Work for clients in North America, Asia & Europe
Michael Schrenk

BIO:

- Minneapolis-based bot writer, consultant & author
- (Soon to be) Las Vegas-based
- Work for clients in North America, Asia & Europe
- Active in my local DEFCON group DC612
BIO:

My DEFCON History

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17

FEAR & HACKING in Las Vegas

An upstanding IS consultant, Michael Schrenk (left) goes to the desert in search of DEF CON depravity. He finds it, and how: “Spot the Fed” contests, rail guns built from scrap, cell phones thrown to the audience like party favors. In Depth, page 75

COMPUTERWORLD
My DEFCON History

Talk:
Introduction to Writing Spiders & Agents
My DEFCON History

Talk:
Online Corporate Intelligence
BIO:

My DEFCON History

Talk:

The Fabulous Executable Image Exploit
Today's Talk:

Screen Scraper Tricks
Difficult Cases

My DEFCON History
Screen Scraper Tricks: Difficult cases

mike@schrenk.com

My book
2007, No Starch Press
San Francisco
Traditional strategies not obsolete

- Downloading, Parsing, Form submission
- Authentication, Stealth, Fault tolerance, etc.

I won't spend a lot of time discussing these things

Supplement traditional approaches with what you learn today
Why are Screen Scrapers Important?

Browsers (alone) are deficient

- Browsers are manual, error prone & time consuming tools
- Browsers do not make decisions for you
- Browsers are not proactive

You won't excel by just doing what everyone else does

- Webbots & Screen scrapers offer competitive advantages
Why are Screen Scrapers Important?

Browsers (alone) are deficient

- Browsers are manual, error prone & time consuming tools
- Browsers do not make decisions for you
- Browsers are not proactive

You won't excel by just doing what everyone else does

Webbots & Screen scrapers offer competitive advantages
Why are Screen Scrapers Important?

Browsers (alone) are deficient

- Browsers are manual, error prone & time consuming tools
- Browsers do not make decisions for you
- Browsers are not proactive

You won't excel by just doing what everyone else does

- Webbots & Screen scrapers offer competitive advantages
Why are Screen Scrapers Important?

Browsers (alone) are deficient

- Browsers are manual, error prone & time consuming tools
- Browsers do not make decisions for you
- Browsers are not proactive

You won't excel by just doing what everyone else does

- Webbots & Screen scrapers offer competitive advantages
Why are Screen Scrapers Important?

Browsers (alone) are deficient
- Browsers are manual, error prone & time consuming tools
- Browsers do not make decisions for you
- Browsers are not proactive

You won't excel by just doing what everyone else does

Webbots & Screen scrapers offer competitive advantages
Review of traditional screen scraping
Review of traditional screen scraping

- Download a web page
Review of traditional screen scraping

• Download a web page
• Manage cookies
Review of traditional screen scraping

• Download a web page
  • Manage cookies
  • Facilitate (SSL) encryption
Review of traditional screen scraping

- Download a web page
  - Manage cookies
  - Facilitate (SSL) encryption
  - Handle server redirection
Review of traditional screen scraping

- Download a web page
- Manage cookies
- Facilitate (SSL) encryption
- Handle server redirection
- Hide your identity with proxies & random timing
Review of traditional screen scraping

- Download a web page
  - Manage cookies
  - Facilitate (SSL) encryption
  - Handle server redirection
  - Hide your identity with proxies & random timing
- Emulate form submission
Review of traditional screen scraping

- Download a web page
  - Manage cookies
  - Facilitate (SSL) encryption
  - Handle server redirection
  - Hide your identity with proxies & random timing
- Emulate form submission
- Parse information from web pages & take action
Review of traditional screen scraping

FREE DOWNLOAD

These tasks (except proxy functions) can be coded with the free PHP code libraries from my book

http://www.schrenk.com/nostarch/webbots/DSP_download.php

- Download a web page
- Manage cookies
- Facilitate (SSL) encryption
- Handle server redirection
- Hide your identity with proxies & random timing
- Emulate form submission
- Parse information from web pages & take action
What constitutes a difficult case?

Either by design—or by accident, web pages have become harder for webbots and screen scrapers to use.
What constitutes a difficult case?

Interstitial web pages

- Commonly used by travel sites when there is a long delay between a database query and a result set.
What constitutes a difficult case?

JavaScript

- When used to *dynamically* modify forms before submission
- Usually solved with my book's online form analyzer.

www.schrenk.com/nostarch/webbots/form_analyzer.php
What constitutes a difficult case?

JavaScript

- AJAX used to populate pages

Example: Expedia

You cannot do a “view source” after first page of search results
What constitutes a difficult case?

Flash
- When used as a navigation technique.

DHTML
- When used as a navigation technique

Elaborate cookie behavior
- Sequence dependent cookies
- Strange JavaScript scripts
What constitutes a difficult case?

Randomly generated form element names

```html
<input
    Type  = "submit"
    Name  = "9S8DUF9S8DUF9S98DFUS9D8FUS9D8FHNSIDJFSIDFJNW983FHSJEFNSKUJFNWO83FJWOSEJKFNSKU3FHS9A38FHIWwe832">
```
FACT: We're still tied to the browser

Sometimes you can fool a server into delivering simpler data formats by pretending to be a mobile device.

Often you need to find a way to emulate browser capability while maintaining full control.
FACT: We're still tied to the browser

Sometimes you can fool a server into delivering simpler data formats by pretending to be a mobile device.

Often you need to find a way to emulate browser capability while maintaining full control.
Browser Macros

- Browser plug-in
Browser Macros

- Browser plug-in
- Readily available
Browser Macros

- Browser plug-in
- Readily available
- Solves all the “Difficult Cases”
Browser Macros

- Browser plug-in
- Readily available
- Solves all the “Difficult Cases”
- Easily extended *(hacked)* beyond intended use
Browser Macros

iMacros solves all of the "difficult cases" because an actual browser is used.

A few additional hacks make it a serious screen scraper tool.
Search for iMacros add-on at addons.mozilla.org

Automate Firefox. Record and replay repetitive work. If you love the Firefox web browser, but are tired of repetitive tasks like visiting the same sites every day, filling out forms, and remembering passwords, then iMacros for Firefox is the solution you've been dreaming of! Whatever you do with Firefox, iMacros can automate it.

Feeds, News & Blogging | Web Development | Privacy & Security | Bookmarks | Social & Communication

Updated June 10, 2009

Download Now
Once iMacros is installed
Start the add-on
And press Record
Screen Scraper Tricks: Difficult cases

mike@schrenk.com

Enter URL
Fill form and press Save

RECORDING A MACRO
Thank you Michael Schrenk for your personal information.
We will be sending you SPAM soon!

Here is your special access code:
13913123123SASDJFSDF2213423423

Please enter the special code above to continue

Start over
Find the \#Current.imm macro

And press “Play”

Your macro will replay!
Switch to demo

This is a REALLY SIMPLE demo!

You need to trust me that it will also work in a much more complex environment (i.e. a “difficult case”)!
### The Macro File (**file_name.iim**)

<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#01</td>
<td>VERSION BUILD=6230608 RECORDER=FX</td>
</tr>
<tr>
<td>#02</td>
<td>TAB T=1</td>
</tr>
<tr>
<td>#03</td>
<td>URL GOTO=<a href="http://www.google.com/">http://www.google.com/</a></td>
</tr>
<tr>
<td>#04</td>
<td>URL GOTO=<a href="http://localhost/defcon17/simple_form.php">http://localhost/defcon17/simple_form.php</a></td>
</tr>
</tbody>
</table>
| #05  | TAG POS=1 TYPE=INPUT:TEXT FORM=NAME:simple_form
     | ATTR=NAME:name CONTENT=Michael Schrenk |
| #06  | TAG POS=1 TYPE=INPUT:TEXT FORM=NAME:simple_form
     | ATTR=NAME:address CONTENT=1725 West Lilac Drive |
| #07  | TAG POS=1 TYPE=INPUT:TEXT FORM=NAME:simple_form
     | ATTR=NAME:city CONTENT=Minneapolis |
| #08  | TAG POS=1 TYPE=INPUT:TEXT FORM=NAME:simple_form
     | ATTR=NAME:state CONTENT=MN |
| #09  | TAG POS=2 TYPE=INPUT:TEXT FORM=NAME:simple_form
     | ATTR=ZIP:state CONTENT=55423 |
| #10  | TAG POS=1 TYPE=INPUT:SUBMIT FORM=NAME:simple_form
     | ATTR=NAME:save&&VALUE:Save |
The Macro File (file_name.iim)

#01 VERSION BUILD=6230608 RECORDER=FX
#02 TAB T=1
#03 URL GOTO=http://www.google.com/
#04 URL GOTO=http://localhost/defcon17/simple_form.php
#05 TAG POS=1 TYPE=INPUT:TEXT FORM=NAME:simple_form
   ATTR=NAME:name CONTENT=Michael<SP>Schrenk
#06 TAG POS=1 TYPE=INPUT:TEXT FORM=NAME:simple_form
   ATTR=NAME:address CONTENT=1725<SP>West<SP>Lilac<SP>Drive
#07 TAG POS=1 TYPE=INPUT:TEXT FORM=NAME:simple_form
   ATTR=NAME:city CONTENT=Minneapolis
#08 TAG POS=1 TYPE=INPUT:TEXT FORM=NAME:simple_form
   ATTR=NAME:state CONTENT=MN
#09 TAG POS=2 TYPE=INPUT:TEXT FORM=NAME:simple_form
   ATTR=ZIP:state CONTENT=55423
#10 TAG POS=1 TYPE=INPUT:SUBMIT FORM=NAME:simple_form
   ATTR=NAME:save&&VALUE:Save

Where Tags can't be identified (FLASH) X/Y coordinates can be used
Dynamic Macro Creation

1. Create a macro Template (text file)
2. Run PHP program to convert template into a macro
3. Run the macro
Creating the Template File

#01  VERSION BUILD=6230608  RECORDER=FX
#02  TAB T=1
#03  URL GOTO=http://www.google.com/
#04  URL GOTO=http://localhost/defcon17/simple_form.php
#05  TAG POS=1  TYPE=INPUT:TEXT  FORM=NAME:simple_form
     ATTR=NAME:name  CONTENT=#_NAME_
#06  TAG POS=1  TYPE=INPUT:TEXT  FORM=NAME:simple_form
     ATTR=NAME:address  CONTENT=#_ADDRESS_
#07  TAG POS=1  TYPE=INPUT:TEXT  FORM=NAME:simple_form
     ATTR=NAME:city  CONTENT=#_CITY_
#08  TAG POS=1  TYPE=INPUT:TEXT  FORM=NAME:simple_form
     ATTR=NAME:state  CONTENT=#_STATE_
#09  TAG POS=2  TYPE=INPUT:TEXT  FORM=NAME:simple_form
     ATTR=NAME:zip  CONTENT=#_ZIP_
#10  TAG POS=1  TYPE=INPUT:SUBMIT  FORM=NAME:simple_form
     ATTR=NAME:save&&VALUE:Save
Substituting Variables

#01  // Get variables (from somewhere, more on this later)
    $name     = (some data)
    $address  = (some data)
    $city     = (some data)
    $state    = (some data)
    $zip      = (some data)

#02  $macro = file_get_contents(“macro.proto”);
#03  $macro = str_replace(“#_NAME_#”, $name, $macro);
#04  $macro = str_replace(“#_ADDRESS_#”, $address, $macro);
#05  $macro = str_replace(“#_CITY_#”, $city, $macro);
#06  $macro = str_replace(“#_STATE_#”, $state, $macro);
#07  $macro = str_replace(“#_ZIP_#”, $zip, $macro);
#08  $macro = file_put_contents(“macro.imm”, $macro);
Substituting Variables

#01  // Get variables (from somewhere, more on this later)
    $name = (some data)
    $address = (some data)
    $city = (some data)
    $state = (some data)
    $zip = (some data)

#02  $macro = file_get_contents("macro.proto");
#03  $macro = str_replace("#_NAME_#", $name, $macro);
#04  $macro = str_replace("#_ADDRESS_#", $address, $macro);
#05  $macro = str_replace("#_CITY_#", $city, $macro);
#06  $macro = str_replace("#_STATE_#", $state, $macro);
#07  $macro = str_replace("#_ZIP_#", $zip, $macro);
#08  $macro = file_put_contents("macro.imm", $macro);
Substituting Variables

#01    // Get variables (from somewhere, more on this later)
$name = (some data)
$address = (some data)
$city = (some data)
$state = (some data)
$zip = (some data)

#02    $macro = file_get_contents(“macro.proto”);
#03    $macro = str_replace(“#_NAME_#”, $name, $macro);
#04    $macro = str_replace(“#_ADDRESS_#”, $address, $macro);
#05    $macro = str_replace(“#_CITY_#”, $city, $macro);
#06    $macro = str_replace(“#_STATE_#”, $state, $macro);
#07    $macro = str_replace(“#_ZIP_#”, $zip, $macro);
#08    $macro = file_put_contents(“macro.imm”, $macro);
Write the Dynamic Macro file

#01 // Get variables (from somewhere, more on this later)
    $name   = (some data)
    $address= (some data)
    $city   = (some data)
    $state  = (some data)
    $zip    = (some data)

#02 $macro = file_get_contents("macro.proto");
#03 $macro = str_replace("#_NAME_#", $name, $macro);
#04 $macro = str_replace("#_ADDRESS_#", $address, $macro);
#05 $macro = str_replace("#_CITY_#", $city, $macro);
#06 $macro = str_replace("#_STATE_#", $state, $macro);
#07 $macro = str_replace("#_ZIP_#", $zip, $macro);
#08 $macro = file_put_contents("macro.imm", $macro);
Write the Dynamic Macro file

#01 // Get variables (from somewhere, more on this later)
$name = (some data)
$address = (some data)
$city = (some data)
$state = (some data)
$zip = (some data)

#02 $macro = file_get_contents("macro.proto");
#03 $macro = str_replace("#_NAME_#", $name, $macro);
#04 $macro = str_replace("#_ADDRESS_#", $address, $macro);
#05 $macro = str_replace("#_CITY_#", $city, $macro);
#06 $macro = str_replace("#_STATE_#", $state, $macro);
#07 $macro = str_replace("#_ZIP_#", $zip, $macro);
#08 $macro = file_put_contents("macro.proto", $macro);

Use this substitution technique to dynamically:
1. Program form field values
2. Change the website URL
3. Change delay times
4. Change destination files
5. Change status message values
6. Etc., etc., etc.
Write the Dynamic Macro file

#01    // Get variables (from somewhere, more on this later)
$name   = (some data)
$address= (some data)
$city   = (some data)
$state  = (some data)
$zip    = (some data)
#02    $macro = file_get_contents("macro.proto");
#03    $macro = str_replace("#_NAME_#", $name, $macro);
#04    $macro = str_replace("#_ADDRESS_#", $address, $macro);
#05    $macro = str_replace("#_CITY_#", $city, $macro);
#06    $macro = str_replace("#_STATE_#", $state, $macro);
#07    $macro = str_replace("#_ZIP_#", $zip, $macro);
#08    $macro = file_put_contents("macro.proto", $macro);

Use the programmability to:

1. Create loops
2. Change data sources
3. Send status messages to central server
4. Etc., etc., etc.
Launching iMacros (macro) from PHP

```php
<?php
if ($os == "linux") {
    system("firefox http://www.google.com");
    sleep(5);
    system("firefox http://run.imacros.net/?m=macro_name.iim");
} else {
    system("start /B firefox http://run.imacros.net/?m=macro_name.iim");
}
?>
```
Launching iMacros (macro) in a cron

I've had better luck launching iMacros (as a scheduled task) as a batch file (Windows) or a BASH file (Linux)

If scheduled on a Linux system, remember to specify a video output.

Display =:0 php /pathname/php_program.php
Launching iMacros (macro) in a cron

I've had better luck launching iMacros (as a scheduled task) as a batch file (Windows) or a BASH file (Linux).

If scheduled on a Linux system, remember to specify a video output.

```
Display=:0 php /pathname/php_program.php
```
iMacros Hints

- **Always** dedicate a browser for iMacros use.
- If you don't use the commercial version of iMacros, use Firefox.
- Make sure that iMacros is activated in the browser before launching a macro.
iMacros Hints

- **Always** dedicate a browser for iMacros use.
- If you don't use the commercial version of iMacros, use Firefox.
- Make sure that iMacros is activated in the browser before launching a macro
iMacros Hints

- **Always** dedicate a browser for iMacros use.
- If you don't use the commercial version of iMacros, use Firefox.
- Make sure that iMacros is activated in the browser before launching a macro.
Preferred iMacco Header commands

#01 '##################################################
#02 ' Set maximum web page time out
#03 SET !TIMEOUT 240
#04 ' Tell iMacros to ignore error messages
#05 SET !ERRORIGNORE YES
#06 ' Clear ALL cookies
#07 CLEAR
#08 ' Initialize Browser tab 1, close all other tabs
#09 TAB T=1
#10 TAB CLOSEALLOTHERS
#11 ' Tell iMacros to ignore images (nice if using Tor)
#12 FILTER TYPE=IMAGES STATUS=ON
#13 ' Tell iMacros to ignore extract messages
#14 SET !EXTRACT_TEST_POPUP NO
#15 '##################################################
Preferred iMaco Header commands

#01    '###################################################################
#02    ' Set maximum web page time out
#03    SET !TIMEOUT 240
#04    ' Tell iMacros to ignore error messages
#05    SET !ERRORIGNORE YES
#06    ' Clear ALL cookies
#07    CLEAR
#08    ' Initialize Browser tab 1, close all other tabs
#09    TAB T=1
#10    TAB CLOSEALLOTHERS
#11    ' Tell iMacros to ignore images (nice if using Tor)
#12    FILTER TYPE=IMAGES STATUS=ON
#13    ' Tell iMacros to ignore extract messages
#14    SET !EXTRACT_TEST_POPUP NO
#15    '###################################################################
Preferred iMacco Header commands

#01  '###########################################################
#02  ' Set maximum web page time out
#03  SET !TIMEOUT 240
#04  ' Tell iMacros to ignore error messages
#05  SET !ERRORIGNORE YES
#06  ' Clear ALL cookies
#07  CLEAR
#08  ' Initialize Browser tab 1, close all other tabs
#09  TAB T=1
#10  TAB CLOSEALLOTHERS
#11  ' Tell iMacros to ignore images (nice if using Tor)
#12  FILTER TYPE=IMAGES STATUS=ON
#13  ' Tell iMacros to ignore extract messages
#14  SET !EXTRACT_TEST_POPUP NO
#15  '###########################################################
Preferred iMaco Header commands

#01  '#########################################################
#02  ' Set maximum web page time out
#03  SET !TIMEOUT 240
#04  ' Tell iMacros to ignore error messages
#05  SET !ERRORIGNORE YES
#06  ' Clear ALL cookies
#07  CLEAR
#08  ' Initialize Browser tab 1, close all other tabs
#09  TAB T=1
#10  TAB CLOSEALLOTHERS
#11  ' Tell iMacros to ignore images (nice if using Tor)
#12  FILTER TYPE=IMAGES STATUS=ON
#13  ' Tell iMacros to ignore extract messages
#14  SET !EXTRACT_TEST_POPUP NO
#15  '#########################################################
Preferred iMaco Header commands

#01 '##################################################
#02 ' Set maximum web page time out
#03 SET !TIMEOUT 240
#04 ' Tell iMacros to ignore error messages
#05 SET !ERRORIGNORE YES
#06 ' Clear ALL cookies
#07 CLEAR
#08 ' Initialize Browser tab 1, close all other tabs
#09 TAB T=1
#10 TAB CLOSEALLOTHERS
#11 ' Tell iMacros to ignore images (nice if using Tor)
#12 FILTER TYPE=IMAGES STATUS=ON
#13 ' Tell iMacros to ignore extract messages
#14 SET !EXTRACT_TEST_POPUP NO
#15 '##################################################
Preferred iMaco Header commands

#01 '##################################################
#02 ' Set maximum web page time out
#03 SET !TIMEOUT 240
#04 ' Tell iMacros to ignore error messages
#05 SET !ERRORIGNORE YES
#06 ' Clear ALL cookies
#07 CLEAR
#08 ' Initialize Browser tab 1, close all other tabs
#09 TAB T=1
#10 TAB CLOSEALLOTHERS
#11 ' Tell iMacros to ignore images (nice if using Tor)
#12 FILTER TYPE=IMAGES STATUS=ON
#13 ' Tell iMacros to ignore extract messages
#14 SET !EXTRACT_TEST_POPUP NO
#15 '##################################################
Preferred iMaco Header commands

#01  '##################################################
#02  ' Set maximum web page time out
#03  SET !TIMEOUT 240
#04  ' Tell iMacros to ignore error messages
#05  SET !ERRORIGNORE YES
#06  ' Clear ALL cookies
#07  CLEAR
#08  ' Initialize Browser tab 1, close all other tabs
#09  TAB T=1
#10  TAB CLOSEALLOTHERS
#11  FILTER TYPE=IMAGES STATUS=ON
#12  ' Tell iMacros to ignore extract messages
#13  SET !EXTRACT_TEST_POPUP NO
#14  '##################################################

A complete iMacros command reference is available at:

wiki.imacros.net/Command_Reference
Let's look at where the data can come from

Firefox/iMacros equipped Harvester (XP, Ubuntu)
Let's look at where the data can come from

**Firefox/iMacros equipped Harvester (XP, Ubuntu)**

Periodically asks for instructions

**Target Website(s)**

**Central Server**
Let's look at where the data can come from

Target Website(s) → Firefox/iMacros equipped Harvester (XP, Ubuntu) → Central Server

Firefox/iMacros equipped Harvester (XP, Ubuntu) periodically asks for instructions and tells Harvester what to do.
Let's look at where the data can come from

Firefox/iMacros equipped Harvester (XP, Ubuntu)

1. Request data
2. Save Screens
3. Parse results

Periodically asks for instructions
Tells Harvester what to do

Target Website(s)

iMacros Macro

Central Server
Let's look at where the data can come from

Firefox/iMacros equipped Harvester (XP, Ubuntu)

1. Request data
2. Save Screens
3. Parse results

Periodically asks for instructions
Tells Harvester what to do
Update central server
Large scale deployment
(challenges traditional thoughts regarding hosting)
Advanced iMacros Hacks

First example was a very straightforward iMacros example

iMacros also some JavaScript-like scripting compatibility (in the paid version)

iMacros has limited parsing and data extraction capability

While solving many problems—without further hacking, iMacros leaves you with many (or most) browser limitations.
Advanced iMacros Hacks

First example was a very straightforward iMacros example

iMacros also some JavaScript-like scripting compatibility (in the paid version)

iMacros has limited parsing and data extraction capability

While solving many problems—without further hacking, iMacros leaves you with many (or most) browser limitations.
Advanced iMacros Hacks

First example was a very straightforward iMacros example.

iMacros also some JavaScript-like scripting compatibility (in the paid version).

iMacros has limited parsing and data extraction capability.

While solving many problems—without further hacking, iMacros leaves you with many (or most) browser limitations.
Advanced iMacros Hacks

First example was a very straightforward iMacros example

iMacros also some JavaScript-like scripting compatibility (in the paid version)

iMacros has limited parsing and data extraction capability

While solving many problems—without further hacking, iMacros leaves you with many (or most) browser limitations.
Suppose you could execute an iMacros macro in one browser tab...
And then open another browser tab to act on the data iMacros downloaded and

- Parse data
- Read/Write to a database
- Pass data back to the iMacros macro
- Or, **anything** else
Advanced iMacros Hacks

Let’s finish our first example.

When we get to this point:

- Create a 2nd tab
- Launch a local php program in Apache
- Parse the web page
- Return the access code
- Complete the form submission in the original tab

Thank you Michael Schrenk for your personal information.
We will be sending you SPAM soon!

Here is your special access code:
13913123123SASDJFSDF2213423423

Please enter the special code above to continue

Start over.
Advanced iMacros Hacks

Let's finish our first example.

When we get to this point:

- Create a 2nd tab
- Launch a local php program in Apache
- Parse the web page
- Return the access code
- Complete the form submission in the original tab

Thank you Michael Schrenk for your personal information.
We will be sending you SPAM soon!

Here is your special access code: 13913123123SASDJFSDF2213423423

Please enter the special code above to continue:

Start over.
Let's finish our first example.

When we get to this point:

- Create a 2\textsuperscript{nd} tab
- Launch a local php program in Apache
- Parse the web page
- Return the access code
- Complete the form submission in the original tab
Let's finish our first example.

When we get to this point:
- Create a 2\textsuperscript{nd} tab
- Launch a local php program in Apache
- Parse the web page
- Return the access code
- Complete the form submission in the original tab
Let's finish our first example.

When we get to this point:

- Create a 2nd tab
- Launch a local php program in Apache
- Parse the web page
- Return the access code
- Complete the form submission in the original tab

Thank you Michael Schrenk for your personal information.
We will be sending you SPAM soon!

Here is your special access code:
13913123123SASDJSDF2213423423

Please enter the special code above to continue

Start over.
Switch to demo #2

You need to trust me that it will also work in a more complex environment (i.e. a “difficult case”)!
This code was added to the original iMacros macro

#01  '# SAVE A COPY OF THE WEBPAGE TO FILE SYSTEM
#02  SAVEAS TYPE=HTM FOLDER=* FILE=PARSE_FILE.html
#03  '# OPEN A NEW TAB FOR THE PARSING SOFTWARE
#04  TAB OPEN
#05  TAB T=2
#06  URL GOTO=http://localhost/defcon17/simple_parse.php
#07  '
#08  '# READ THE PARSED RESULTS
#09  TAB T=1
#10  CMDLINE !DATASOURCE data.csv
#11  SET !DATASOURCE_COLUMNS
#12  SET !DATASOURCE_LINE {{!LOOP}}
#13  TAG POS=1 TYPE=INPUT:TEXT
   FORM=NAME:simple_form
   ATTR=NAME:access_code CONTENT={{!COL1}}
#14  WAIT SECONDS=5
#15  TAG POS=1 TYPE=INPUT:SUBMIT FORM=NAME:simple_form
   ATTR=NAME:save&&VALUE:Save

Saves a copy of the screen data to a file in the /iMacros/Downloads directory.
This code was added to the original iMacros macro

```
#01  '# SAVE A COPY OF THE WEBPAGE TO FILE SYSTEM
#02  SAVEAS TYPE=HTM FOLDER=* FILE=PARSE_FILE.html
#03  '# OPEN A NEW TAB FOR THE PARSING SOFTWARE
#04  TAB OPEN
#05  TAB T=2
#06  URL GOTO=http://localhost/defcon17/simple_parse.php
#07  '
#08  '# READ THE PARSED RESULTS
#09  TAB T=1
#10  CMDLINE !DATASOURCE data.csv
#11  SET !DATASOURCE_COLUMNS 1
#12  SET !DATASOURCE_LINE {{!LOOP}}
#13  TAG POS=1 TYPE=INPUT:TEXT FORM=NAME:simple_form
    ATTR=NAME:access_code CONTENT={{!COL1}}
#14  WAIT SECONDS=5
#15  TAG POS=1 TYPE=INPUT:SUBMIT FORM=NAME:simple_form ATTR=NAME:save&&VALUE:Save
```

- Opens the second tab
- Loads and runs the file “simple_parse.php” on a local installation of Apache

This program
- Reads the previously stored file
- Parses the access code
- Stores it in a iMacros (CSV) data file
This code was added to the original iMacros macro:

```
#01  '# SAVE A COPY OF THE WEBPAGE TO FILE SYSTEM
#02  SAVEAS TYPE=HTM FOLDER=* FILE=PARSE_FILE.html
#03  '# OPEN A NEW TAB FOR THE PARSE SOFTWARE
#04  TAB OPEN
#05  TAB T=2
#06  URL GOTO=http://localhost/defcon17/simple_parse.php
#07  '
#08  '# READ THE PARSED RESULTS
#09  TAB T=1
#10  CMDLINE !DATASOURCE data.csv
#11  SET !DATASOURCE_COLUMNS 1
#12  SET !DATASOURCE_LINE {{!LOOP}}
#13  TAG POS=1 TYPE=INPUT:TEXT
    FORM=NAME:simple_form
    ATTR=NAME:access_code CONTENT={{!COL1}}
#14  WAIT SECONDS=5
#15  TAG POS=1 TYPE=INPUT:SUBMIT FORM=NAME:simple_form
    ATTR=NAME:save&&VALUE:Save
```

- Return to first tab
- Read (CSV) data file
- Insert data into form

This is a simplified example, can also employ loops (CSV rows) and many more data fields (CSV columns).
This code was added to the original iMacros macro

```
#01  '# SAVE A COPY OF THE WEBPAGE TO FILE SYSTEM
#02  SAVEAS TYPE=HTM FOLDER=* FILE=PARSE_FILE.html
#03  '# OPEN A NEW TAB FOR THE PARSING SOFTWARE
#04  TAB OPEN
#05  TAB T=2
#06  URL GOTO=http://localhost/defcon17/simple_parse.php
#07  '
#08  '# READ THE PARSED RESULTS
#09  TAB T=1
#10  CMDLINE !DATASOURCE data.csv
#11  SET !DATASOURCE_COLUMNS 1
#12  SET !DATASOURCE_LINE {{!LOOP}}
#13  TAG POS=1 TYPE=INPUT:TEXT
    FORM=NAME:simple_form
    ATTR=NAME:access_code CONTENT={{!COL1}}
#14  WAIT SECONDS=5
#15  TAG POS=1 TYPE=INPUT:SUBMIT FORM=NAME:simple_form
    ATTR=NAME:save&&VALUE:Save
```

Submit form
Using **additional tabs** to run **local programs** facilitates advanced features not possible in traditional iMacros configurations.

Interrupted macros
- Parse data from pages and act on results
- Interface with local peripherals
- Change proxy settings
- Aggregate data from multiple websites
- Aggregate services from multiple websites
- Upload data in mid-macro
- Etc., etc., etc.
Using **additional tabs** to run **local programs** facilitates advanced features not possible in traditional iMacros configurations.

**Interrupted macros**

- Parse data from pages and act on results
- Interface with local peripherals
- Change proxy settings
- Aggregate data from multiple websites
- Aggregate services from multiple websites
- Upload data in mid-macro
- Etc., etc., etc.
Using **additional tabs** to run **local programs** facilitates advanced features not possible in traditional iMacros configurations.

**Interrupted macros**
- Parse data from pages and act on results
- Interface with local peripherals
- Change proxy settings
- Aggregate data from multiple websites
- Aggregate services from multiple websites
- Upload data in mid-macro
- Etc., etc., etc.
Using **additional tabs** to run **local programs** facilitates advanced features not possible in traditional iMacros configurations.

**Interrupted macros**
- Parse data from pages and act on results
- Interface with local peripherals
- Change proxy settings
- Aggregate data from multiple websites
- Aggregate services from multiple websites
- Upload data in mid-macro
- Etc., etc., etc.
Using additional tabs to run local programs facilitates advanced features not possible in traditional iMacros configurations.

Interrupted macros
- Parse data from pages and act on results
- Interface with local peripherals
- Change proxy settings
- Aggregate data from multiple websites
- Aggregate services from multiple websites
- Upload data in mid-macro
- Etc., etc., etc.
Using **additional tabs** to run **local programs** facilitates advanced features not possible in traditional iMacros configurations.

**Interrupted macros**
- Parse data from pages and act on results
- Interface with local peripherals
- Change proxy settings
- Aggregate data from multiple websites
- Aggregate services from multiple websites
- Upload data in mid-macro
- Etc., etc., etc.
Using additional tabs to run local programs facilitates advanced features not possible in traditional iMacros configurations.

Interrupted macros
- Parse data from pages and act on results
- Interface with local peripherals
- Change proxy settings
- Aggregate data from multiple websites
- Aggregate services from multiple websites
- Upload data in mid-macro
- Etc., etc., etc.
Using **additional tabs** to run **local programs** facilitates advanced features not possible in traditional iMacros configurations.

**Interrupted macros**

- Parse data from pages and act on results
- Interface with local peripherals
- Change proxy settings
- Aggregate data from multiple websites
- Aggregate services from multiple websites
- Upload data in mid-macro
- Etc., etc., etc.
Using **additional tabs** to run **local programs** facilitates advanced features not possible in traditional iMacros configurations.

**Interrupted macros**

- Parse data from pages and act on results
- Interface with local peripherals
- Change proxy settings
- Aggregate data from multiple websites
- Aggregate services from multiple websites
- Upload data in mid-macro
- Etc., etc., etc.
Heartwarming moment
ReCAPTCHA

250 million CAPTCHAS executed daily
Free CAPTCHA service
30 million of these CAPTCHAS are solved daily
CAPTCHA words are scanned from old manuscripts
Solved CAPTCHAS actually digitize manuscripts
ReCaptcha

250 million CAPTCHAs executed daily
Free CAPTCHA service
30 million of these CAPTCHAs are solved daily
CAPTCHA words are scanned from old manuscripts
Solved CAPTCHAs actually digitize manuscripts
ReCAPTCHA

250 million CAPTCHAS executed daily
Free CAPTCHA service
30 million of these CAPTCHAS are solved daily
CAPTCHA words are scanned from old manuscripts
Solved CAPTCHAS actually digitize manuscripts
ReCAPTCHA

250 million CAPTCHAS executed daily
Free CAPTCHA service
30 million of these CAPTCHAS are solved daily
CAPTCHA words are scanned from old manuscripts
Solved CAPTCHAS actually digitize manuscripts
ReCAPTCHA

250 million CAPTCHAS executed daily
Free CAPTCHA service
30 million of these CAPTCHAS are solved daily
CAPTCHA words are scanned from old manuscripts
Solved CAPTCHAS actually digitize manuscripts
ReCAPTCHA

250 million CAPTCHAS executed daily
Free CAPTCHA service
30 million of these CAPTCHAS are solved daily
CAPTCHA words are scanned from old manuscripts
Solved CAPTCHAS actually digitize manuscripts
The Breckinridge and Lane Democrats, having taken courage at the recent eastern advices, are organizing energetically for the campaign. Several prominent Democrats who at first favored Douglas, are coming out for the other side, apparently under the pressure of Federal influence. An address to the National
CAPTCHA Solving Services (APIs)

There are services (APIs) that solve CAPTCHAs.
CAPTCHA Solving Services (APIs)

There are services (APIs) that solve CAPTCHAs.

Unlike OCR, these are solved by REAL people.
There are services (APIs) that solve CAPTCHAs. Unlike OCR, these are solved by REAL people. Do a quick Google search for details.
Heartwarming moment

There are CAPTCHA solving services
Heartwarming moment

There are CAPTCHA solving services

1. CAPTCHA DISPLAYED ON WEB PAGE
2. CAPTCHA IMAGE SENT TO SERVICE
Heartwarming moment

There are CAPTCHA solving services

- CAPTCHA DISPLAYED ON WEB PAGE
- CAPTCHA IMAGE SENT TO SERVICE
- CAPTCHA SOLVED BY HUMAN
Heartwarming moment

There are CAPTCHA solving services

1. CAPTCHA DISPLAYED ON WEB PAGE
2. CAPTCHA IMAGE SENT TO SERVICE
3. CAPTCHA SOLVED BY HUMAN
4. EMBEDDED TEXT SENT BACK TO REQUESTOR
Heartwarming moment

There are CAPTCHA solving services

1. CAPTCHA DISPLAYED ON WEB PAGE
2. CAPTCHA IMAGE SENT TO SERVICE
3. CAPTCHA SOLVED BY HUMAN
4. EMBEDDED TEXT SENT BACK TO REQUESTOR
5. TEXT IS ENTERED IN CAPTCHA TEXTBOX
Heartwarming moment

There are CAPTCHA solving services

- CAPTCHA DISPLAYED ON WEB PAGE
- CAPTCHA IMAGE SENT TO SERVICE
- CAPTCHA SOLVED BY HUMAN
- EMBEDDED TEXT SENT BACK TO REQUESTOR
- TEXT IS ENTERED IN CAPTCHA TEXTBOX
- CAPTCHA SOLVED! (Unintentional Consequences)
There are CAPTCHA solving services.

**Heartwarming moment**

**A FEEL GOOD WIN-WIN SITUATION!**

There are CAPTCHA solving services.

Spammers pay to digitize old documents.

People in developing nations have jobs.

CAPTCHA displayed on web page.

CAPTCHA image sent to service.

CAPTCHA solved by human.

Text solved.

Text is entered in CAPTCHA text box.

CAPTCHA solved!

(Unintentional consequences)

EMBEDDED TEXT SENT BACK TO REQUESTOR.

A feel good win-win situation!
In conclusion

Review of traditional scraper theory
Described web design technologies and techniques that create “difficult cases” for webbot/screen scraper developers
Saw that iMacros can solve most (all) difficult cases by:
  Absolute browser emulation
  Complete control (through hacks)
Looked at managing large scale deployments
In conclusion

Review of traditional scraper theory
Described web design technologies and techniques that create “difficult cases” for webbot/screen scraper developers
Saw that iMacros can solve most (all) difficult cases by:
  - Absolute browser emulation
  - Complete control (through hacks)
Looked at managing large scale deployments
In conclusion

Review of traditional scraper theory
Described web design technologies and techniques that create “difficult cases” for webbot/screen scraper developers
Saw that iMacros can solve most (all) difficult cases by:
  Absolute browser emulation
  Complete control (through hacks)
Looked at managing large scale deployments
In conclusion

Review of traditional scraper theory
Described web design technologies and techniques that create “difficult cases” for webbot/screen scraper developers
Saw that iMacros can solve most (all) difficult cases by:
   Absolute browser emulation
   Complete control (through hacks)
Looked at managing large scale deployments
In conclusion

Review of traditional scraper theory
Described web design technologies and techniques that create “difficult cases” for webbot/screen scraper developers
Saw that iMacros can solve most (all) difficult cases by:
  Absolute browser emulation
  Complete control (through hacks)
Looked at managing large scale deployments
In conclusion

Review of traditional scraper theory
Described web design technologies and techniques that create “difficult cases” for webbot/screen scraper developers
Saw that iMacros can solve most (all) difficult cases by:
  - Absolute browser emulation
  - Complete control (through hacks)
Looked at managing large scale deployments
Thank you!

Questions?

www.schrenk.com
mike@schrenk.com
twitter.com/mgschrenk