

Unmanned Aerial Vehicles



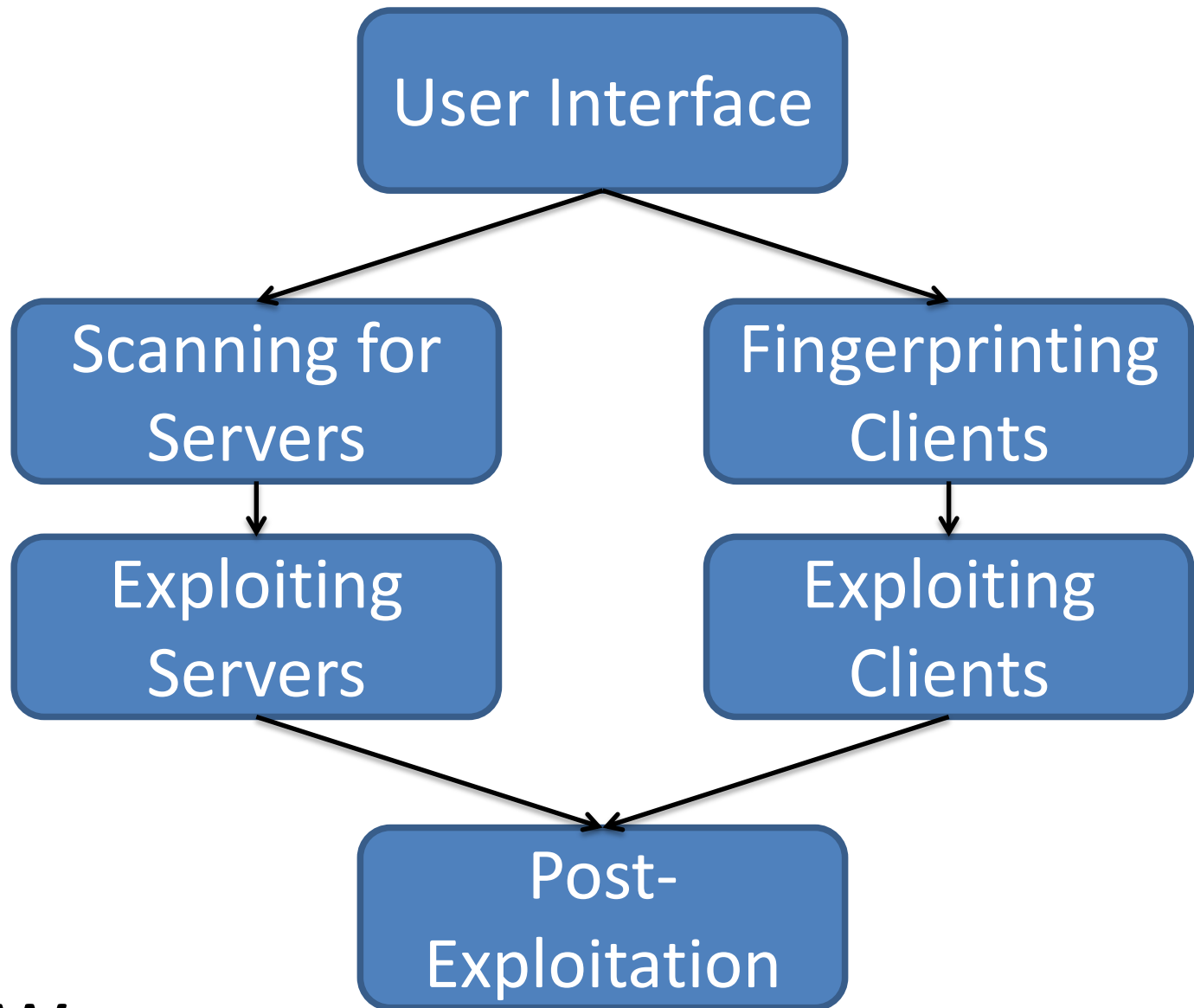
Exploit Automation with
the Metasploit Framework

James Lee

whoami

- James Lee
- egypt
- Core Developer, Metasploit Project
- Working full time on Metasploit for





Overview



Automating msfconsole

- Resource files
 - A list of commands to be run in sequence
 - Can be anything you would type at the msf> prompt
- setg
- save



Resource files

- `$./msfconsole -r foo.rc`
- `msf> resource foo.rc`
- `~/.msf3/msfconsole.rc`
 - Loaded on startup



Example Resource File

```
setg RHOSTS 10.1.1.1-254
```

```
setg USERNAME Administrator
```

```
setg PASSWORD password
```

```
use auxiliary/scanner/smb/smb_login
```

```
run
```

```
use auxiliary/scanner/telnet/telnet_login
```

```
run
```



ORACLE®



SERVERS



Scanning

- Have to **find** servers before you can exploit them
- Metasploit has several ways to do this
 - Run nmap and nexpose directly from the console
 - Import other tools' output
 - MSF built-in scanners (auxiliary/scanner/*)



Israeli Orbiter, surveillance UAV





- Two options:
 - Run nmap normally with -oX and use db_import to store the results
 - db_nmap command will run nmap and handle the import for you
- Either way, results get stored in the database



- nexpose_scan
- db_import
- If you have a Community license (free), limited to 32 IP addresses at a time
 - Msf will scan the whole range in 32-address chunks

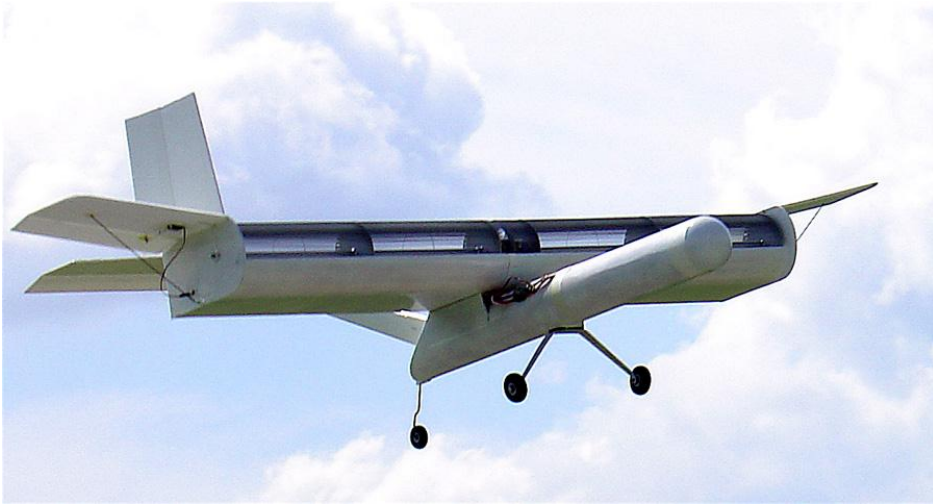
```
msf > load nexpose  
  
NeXpose  
  
Screenshot-1.png  
  
[*] NeXpose integration has been activated  
[*] Successfully loaded plugin: nexpose  
msf > 
```

Nexpose

- Also stores vulnerability references
 - CVE, BID, ...
 - Without these, figuring out which exploits to run can be more difficult
- Can be used to launch exploits as well



MSF Built-in Scanning



FanWing Surveillance Platform

- Implemented as auxiliary modules
- Aux is like an exploit without a payload
 - Usage similar to exploits
- Can go through meterpreter routes



Faster Setup

- RHOSTS can be nmap-notation or “file:<filename>”
- File should contain nmap-notation address ranges
 - e.g.:
 - 10.1.1.2,5,7-254
 - 10.2.2.*
 - 10.3.3.0/24



Faster Scanning

- set THREADS 256
 - Windows freaks out after 16 threads
 - Cygwin doesn't handle more than about 200
 - Linux? Go to town.
- Caveat: tunneling through meterpreter



Selected Scanners

- Informational
 - smb_version
 - netbios/nbname
- Pwnage
 - smb_login
 - telnet_login
 - mssql_login
 - vnc_none_auth



Server Exploits

- The bulk of msf's exploit modules
 - 385 as of Jan 9
- Many protocols implemented in an exploit-friendly way
 - smtp, imap, http, smb, dcerpc, sunrpc, ftp, ...
- Wide range of protocol-level IDS evasions



Automatically Exploiting Servers

- db_autopwn
- NeXpose plugin



db_autopwn

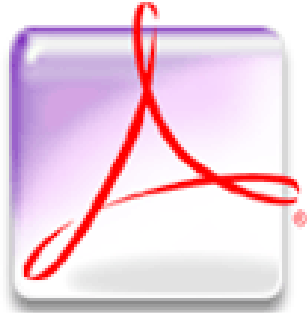
- Need to have targets stored in the db
- If vulnerability references are available, can cross-reference against specific hosts
- Can just use matching ports if you don't have refs
- Checks global MinimumRank to limit exploits to a particular safety level



NeXpose

- Scan, detect, exploit all in one command
 - `nexpose_scan -x <host range>`
 1. Populates the db with hosts, services, vulns
 2. Cross-references vulns and exploits
 3. Throws exploits at vulnerable servers
- Has the potential to give you tons of shells
- Can take a long time for lots of hosts
- Uses MinimumRank as well





CLIENTS



Client Fingerprinting



- User Agent

- Easy to spoof
- Easy to change in a proxy
- Some third-party software changes it
- Less often changed in JavaScript



Fingerprinting the Client

- Various JS objects only exist in one browser
 - `window.opera`, `Array.every`
- Some only exist in certain versions
 - `window.createPopup`, `Array.every`, `window.Iterator`
- Rendering differences and parser bugs
 - IE's conditional comments



Internet Explorer

- Parser bugs, conditional comments
 - Reliable, but not precise
- ScriptEngine*Version()
 - Almost unique across all combinations of client and OS, including service pack
- ClientCaps



Opera

- `window.opera.version()`
 - Includes minor version, e.g. “9.61”
- `window.opera.buildNumber()`
 - Different on each platform for a given version
 - e.g.: “8501” == Windows
 - Not precise, only gives platform, no version or service pack



Hybrid Approach for FF

- Existence of `document.getElementsByClassName` means Firefox 3.0
- If UA says IE6, go with FF 3.0
- If UA says FF 3.0.8, it's probably not lying, so use the more specific value



Firefox OS Detection

- Most of the objects used in standard detection scripts are affected by the User-Agent
 - E.g., when spoofing as iPhone, `navigator.platform = "iPhone"`
- `navigator.oscpu` is not
 - "Linux i686"
 - "Windows NT 6.0"



Safari / Webkit

- Infuriatingly standards compliant in JS
- Can detect its existence easily
 - `window.WebkitPoint`, many others
- Most Safari-specific stuff has been around since 1.2, so not useful for version detection



Chrome / Webkit

- Same javascript engine as Safari
- So far, no easy way to change UA
- navigator.vendor is always “Google Inc.”



Client Exploits in MSF

- Extensive HTTP support
 - Heapspray in two lines of code
 - Sotirov's .NET DLL, heap feng shui
- Wide range of protocol-level IDS evasion
- Simple exploit in ~10 lines of code



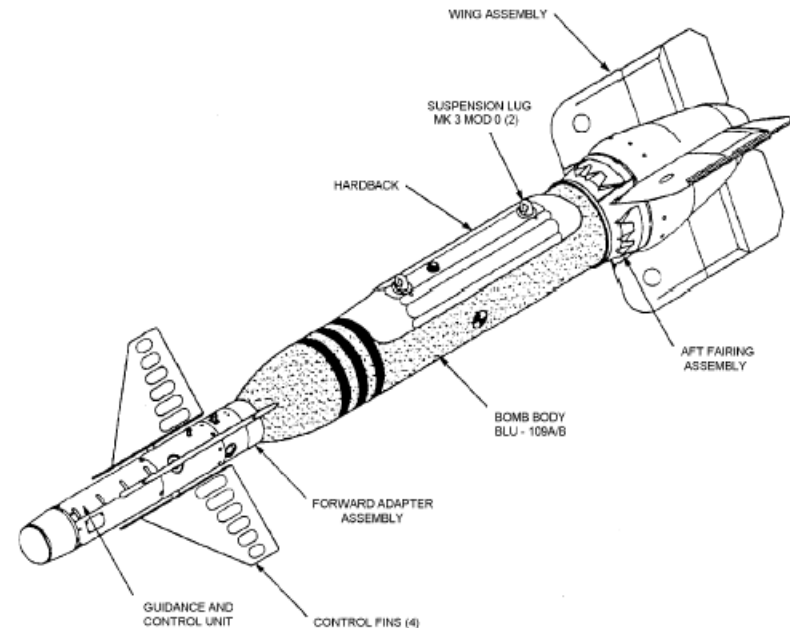
Automatically Exploiting Clients

- Browser Autopwn Auxiliary module
 - I spoke about this at Defcon in 2009
- Fingerprints a client
 - Stores detection in the database
- Determines what exploits might work
 - Uses MinimumRank, too
- Tries the ones most likely to succeed



Advantages of Browser Autopwn

- OS and client detection is client-side, more reliable in presence of spoofed or broken UA
- Detection results automatically stored in the database
- Not written in PHP
 - PHP sucks



Browser Autopwn Usage

```
msf> use auxiliary/server/browser_autopwn
msf (browser_autopwn)> set URIPATH /
msf (browser_autopwn)> set EXCLUDE opera
msf (browser_autopwn)> set MATCH .*
msf (browser_autopwn)> run

[*] Starting exploit modules on host 10.1.1.1...
[*] ---
```



Automating Users

- Browser Autopwn automates the exploits but how do we get users to come to our evil web server?



Karmetasploit

- Wireless Access Point of Doom
- Using aircrack-ng, appears to be every access point that anybody probes for
 - “Why, yes, I am Office_WiFi, please connect”
- Lets you control the route, the DNS, everything
 - “Yup, I'm your internal web server. And your email server. And your file server. And...”



More on Karma

- Actually about 5 years old
- It still works amazingly well
- More info about getting it working is on our wiki:
<http://www.metasploit.com/redmine/projects/framework/wiki/Karmetasploit>



Assagai

- Complete phishing framework
- Uses Metasploit exploits and payloads
- Gathers other statistics
- Has common email templates



Welcome to Assagai

Assagai is a framework designed to simplify the process of creating and executing phishing exercises for penetration testing engagements.

Assagai allows for the creation of custom emails, websites and payloads [pdf, zip, doc] for targeting organizations. It will generate reports showing trends, success rate and other metrics as well as captured data such as usernames, passwords, system information, plugins and more...

Create New Phishing Engagement

Name:

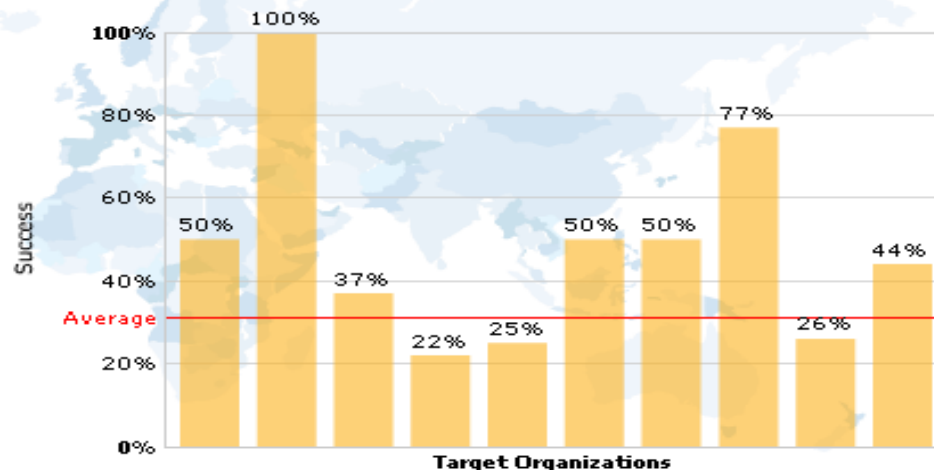
Description:

Copy Existing Phishing Engagement











This makes a copy of the selected phish including all the email targets that were originally uploaded.

Name:

Phishing Engagement Success Rate
[% of users clicking link]



Previous Phishing Engagement Results

Date	Track	Engagement	Description	Success Rate
2009-10-07 23:21:34		Demo_all	this is just another demo of assagai and it's functionality.	 50%
2009-10-07 20:21:22		Demo_mailer	This ia test of the mail sending status page	 100%
2009-09-18 19:38:25		new_mailer_test	This is a test of the new mailer .php code	 50%
2009-07-18 09:14:33		performance	a test of the database performance	 77%
2009-07-09 23:28:27		test_plugins	test of plugin writing to db	 44%

[First](#) [Previous](#)

[Home](#)[Manage](#)[Reports](#)[About](#)

Select the email and webpage template and theme to use for your phish. You will be able to edit the body of the email and the contents of the webpage in the next steps. Alternatively you can create a custom email or use the mirroring option to scrape a target website.

In order to permanently add a template to the current list of available templates please click on [manage] above.

Select Email Theme

- Corporate Compliance**

This email attempts to take advantage of the various compliance and regulatory requirements that companies have today.
- Password Synchronization**

This email attempts to convince a user that by 'syncing' their password, they will be able to sign into all their applications with one login.
- IRS Refund**

This phish attempts to take advantage of tax season and hints at a user getting an additional refund
- Watchguard SSLVPN**

This phish attempts to take advantage of an update email about a SSLVPN service
- Human Resources Benefits Update**

This email attempts to convince the user that they need to login to the Human Resources Portal

Select Webpage Theme

- Corporate Compliance**

This webpage attempts to take advantage of the various compliance and regulatory requirements that companies have.
- Password Synchronization**

This webpage attempts to convince a user that by 'syncing' their password, they will be able to sign into all their applications with one login.
- IRS Refund**

This website attempts to take advantage of tax season and hints at a user getting an additional refund
- Watchguard SSLVPN**

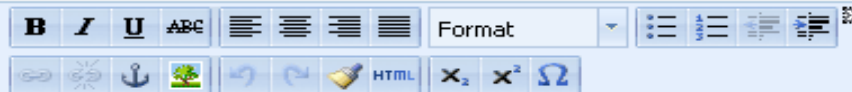
This website presents a fake SSLVPN login page and asks the user to download an updated client
- Human Resources Benefits Manager**

This template presents a fake login page to the company's online benefits manager

This page allows you to edit and make changes to the selected website phishing theme. You will be able to edit the body and the contents of the webpage, adding variables to insert iframes for plugin detection, inline pdf's, usernames, first name & last name fields and urls.

Edit Web Phish Theme:

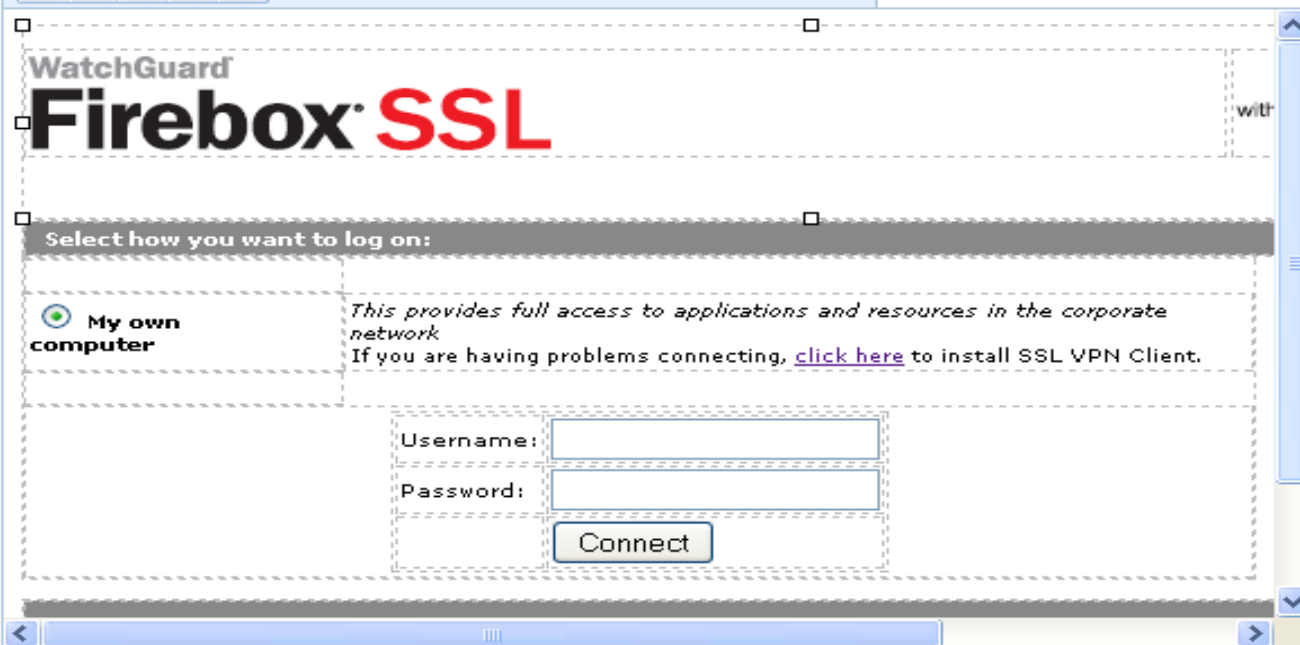
Selected Theme:



Description:

Page Title:

Page Body:



The body of the website needs to be well formatted html. A series of variables are available that can be added to the body in order to personalize it. Please check the help file for a listing of these variables and examples of their usage.

[Continue](#)

Metaphish

- Use the target's public information against them
- See valsmith, Colin, and dkerb's talk from BH USA 2009



Automating Post-exploitation

- Meterpreter scripts
 - set AutoRunScript <script name>
- Plugins
 - Can be auto loaded at startup with resource files



Meterpreter scripts

- Just a ruby script
- Easy to write, lots of flexibility
- Access to Meterpreter API



Meterpreter API

- Core + Extensions
 - Core is basic, mostly useful for loading extensions
- Current extensions:
 - Stdapi
 - Priv, Incognito
 - Espia
 - Sniffer



Meterpreter Stdapi: process

- `client.sys.process`
 - Acts like a Hash, where keys are image names and values are process IDs
 - `client.sys.process['explorer.exe']`
 - `=> 1408`



Meterpreter Stdapi: memory

```
p =  
client.sys.process.open(pid,PROCESS_ALL_ACCESS)  
addr = p.memory.allocate(length)  
p.memory.write(addr, "stuff")  
p.thread.create(addr)
```



Meterpreter Stdapi: filesystem

- `client.fs.file.upload_file(dest, source)`
- `client.fs.file.download_file(dest, source)`
- `client.fs.file.expand_path("%TEMP%")`



Priv and Incognito

- Stuff that requires privileges, SYSTEM preferred
- Priv
 - Dump hashes, alter file MACE
- Incognito
 - list impersonation/delegation tokens



Espia

- `client.espia.espia_image_get_dev_screen`
 - Returns a bitmap as a String
 - From commandline, 'screenshot' stores to file
- `client.espia.espia_audio_get_dev_audio`
 - No command for this yet, only available from API



Meterpreter Sniffer

- `client.sniffer.capture_start`
 - Starts capturing
- `client.sniffer.capture_dump`
 - Puts the captured packets into a buffer we can read
- `client.sniffer.capture_dump_read`
 - Reads from the buffer



Sniffer caveat

- The packet format isn't standard, so we have to convert it to PCAP to be useful
- Console command does it for you



Some Nifty Existing Scripts

- `vnc` -- Uploads a VNC server to the target and tunnels traffic through the current TCP connection or a new connect-back
- `packetrecorder` -- Starts a sniffer on the target and retrieves packets every `<interval>` seconds
- `persistence` -- Builds a `meterpreter.exe` that connects back every `<interval>` seconds
- `killav` -- Runs through a list of known Anti-Virus process names and kills anything that matches



Colin and Dave's talk

- Don't miss it
- Right after lunch
- About using meterpreter's memory API for doing all kinds of crazy stuff



MSF Plugins

- Can extend or replace parts of the framework
- Full access to Rex and Msf APIs
- Can add callbacks for various events, add commands to the console, anything you can think of



Hooking sessions from a plugin

```
include SessionEvent  
  
def on_session_open(session)  
    # Do something with the session  
end  
  
def initialize(framework, opts)  
    framework.events.add_session_subscriber(self)  
end
```



Some notable events

- `on_session_open`
- `on_module_run`
- `on_exploit_success`



Some Nifty Existing Plugins

- `db_credcollect` – automatically retrieves hashes from new meterpreter sessions, stores them in the database
- `pcap_log` – just like running `tcpdump` in the background
- `session_tagger` – creates a directory on new sessions as proof of compromise



Demonstrations



Conclusions

- Lots of automation available that requires no programming skills
- A little bit of ruby gives you lots of power and flexibility
- Don't type any more than you have to
 - Carpal Tunnel Syndrome sucks



Download it

- `svn co http://metasploit.com/svn/framework3/trunk`
- Submit patches to `msfdev@metasploit.com`



Questions?

