

**metasploit**

# who am i ?

H D Moore <hdm [at] metasploit.com>

**metasploit PROJECT**

Core developer and project lead

**BreakingPoint Systems**

Director of Security Research

# why listen ?

- A great tool you can use today
- The BSD-licensed Rex library
- Latest in exploit technology

# metasploit framework

- An exploit development platform
  - Security researchers
  - Penetration testers
  - Security vendors
  - Script kiddies

# metasploit history

- Version 1.0 (2003-2004)
  - Perl, 15 exploits, curses UI
- Version 2.7 (2003-2006)
  - Perl, 150+ exploits, 3 Uis
- Version 3.0 (2007+)

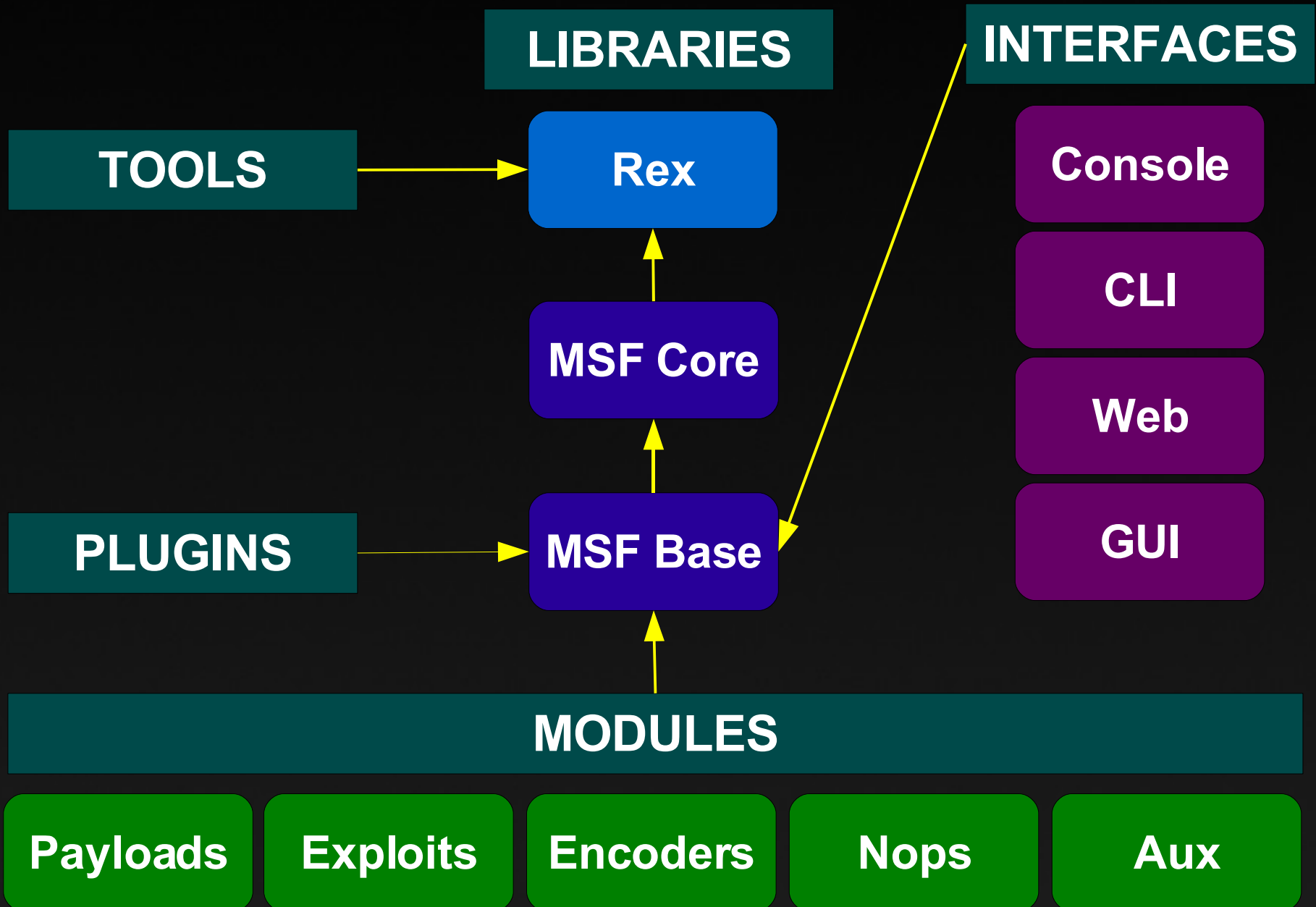
# metasploit 3.0

- 100,000 lines of Ruby
- 53,000 lines of C/C++
- 8000 lines of ASM
- 350 unique modules
- 2 years to develop

# why ruby ?

- Clean, easy, and fun
- Awesome OO model
- Green threading
- Platform support

# architecture



# the Rex library

- Text manipulation
- CPU instructions
- Fancy sockets
- File formats
- Protocols
  - SMB, DCERPC, SUNRPC, HTTP

# metasploit modules

- Simple Ruby classes
- Dynamically loaded
- Rich meta-information
- Expose type-specific methods

# metasploit modules

- Attacks come in two flavors
  - Exploits
  - Auxiliary
- Exploits require other modules
  - Payload, Encoder, Nops

# metasploit exploits

- Exploits trigger the vulnerability
  - Buffer overflow, format string, etc
- Payloads determine what happens
  - Remote shell ( # )
  - Execute command
  - Add user account

# metasploit exploits

- Encoders transform the payload
  - Remove restricted characters
  - Help with IDS/IPS evasion
- Nops are used for padding
  - Pad the payload buffer to static size
  - Helps with IDS/IPS evasion too! :)

# metasploit modules

- Simple Ruby classes
- Dynamically loaded
- Rich meta-information
- Expose type-specific methods

# metasploit exploits

- Modules inherit Msf::Exploit
- Heavy use of Ruby mixins
  - TCP, UDP, SMB, HTTP
  - Active, Passive, Brute force
  - WiFi, Pcap, Bluetooth

# exploit example

connect

```
print_status("Trying target #{target.name}...")
```

```
buf = Rex::Text.rand_text_english(1816)
```

```
seh = generate_seh_payload(target.ret)
```

```
buf[1008, seh.length] = seh
```

```
send_cmd( ['USER', buf] , false)
```

handler

disconnect

# metasploit payloads

- Modules inherit Msf::Payload
- Singles, Stagers, Stages
  - Remote command shells
  - In-memory DLL injection
  - “CMD” payload types
  - “PHP” payload types

# windows payloads

- Standardized calling convention
- Tiny payloads via ordinal resolution
- DLL injection payloads
  - In-memory VNC server
  - PassiveX payload stager
  - The Meterpreter...

# the meterpreter

- Dynamically extensible payload
- Custom network protocol
- The basic “stdapi” extension
  - ps, kill, ls, rm, mkdir, rmdir
  - upload, download, execute
  - migrate, interact, load, scripting

# metasploit auxiliaries

- Modules inherit Msf::Auxiliary
- Anything not an “exploit”
  - Discovery and fingerprinting
  - Network protocol “fuzzers”
  - Denial of service methods
  - Administrative access exploits

# **user interfaces**

- **msfconsole**
- **msfcli**
- **msfweb**
- **msfgui**

# events

- Registered subscriber model
- Trigger on common actions
  - Exploit launched
  - Session creation
  - Job creation
  - User command

# plugins

- Hook events, extend objects
- Examples
  - Socket filtering and logging
  - Database support
  - Exploit automation
  - Telnet console

# summary

- An advanced exploit toolkit
- Simple to use and extend
- 3.0 is stable, 3.1 is coming SOON!

<http://framework.metasploit.com/>

**demos !**