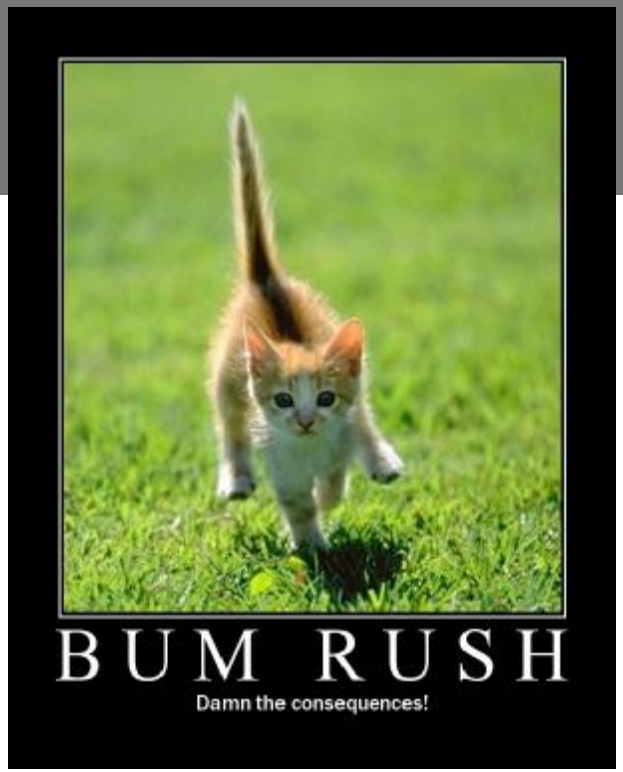
Advances in HTTP encapsulated payloads
Or, a Young Metasploit User's Illustrated Primer

DC405 03/20/2009

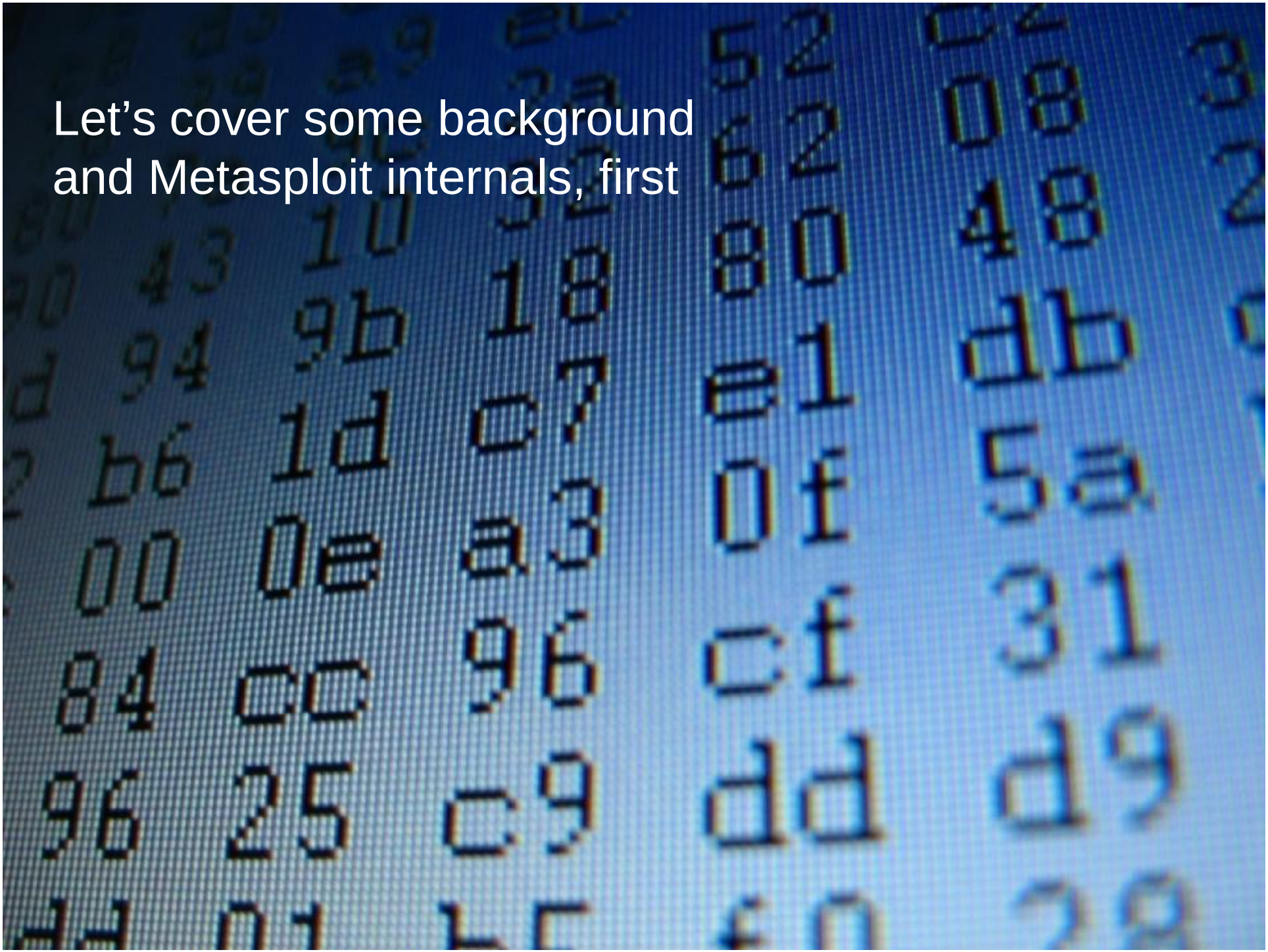
"4444 is the new 31337"

Nathan Keltner
natron [at] metasploit.com
http://blog.invisibledenizen.org

This is the hill we're storming tonight, kiddos:



- metasploit overview
  - structure / design / etc
  - payload connect methods
- Internet Explorer overview
  - security settings / implementation
- PassiveX payloads / reverse_http / "http encapsulated payloads"


Let’s cover some background
and Metasploit internals, first

WTF is metasploit? I present to you a death-by-bullets brain dump of random facts.

- Community driven, open source (as of v3.2)
- Exploit development framework
- Designed for research and testing, but useful in PT
- 300+ exploits, 200+ payloads
- Windows, Linux, Mac OS X, BSD, Solaris, AIX, IRIX
- Wifi driver exploits + kernel payloads
- msfpayload, metasm, msfgui, scruby, byakugan
- Ridiculous payload encoding
- Tons of brand-spanking-new Mac OS X code
- Largest Ruby project currently in existence

On a complete side note but speaking of Apple, for ri0t, 0hm, jayson, kin, and all you other fanboys:

"The things that Windows do to make it harder, Macs don't do. Hacking into Macs is so much easier. You don't have to jump through hoops and deal with all the anti-exploit mitigations you'd find in Windows.

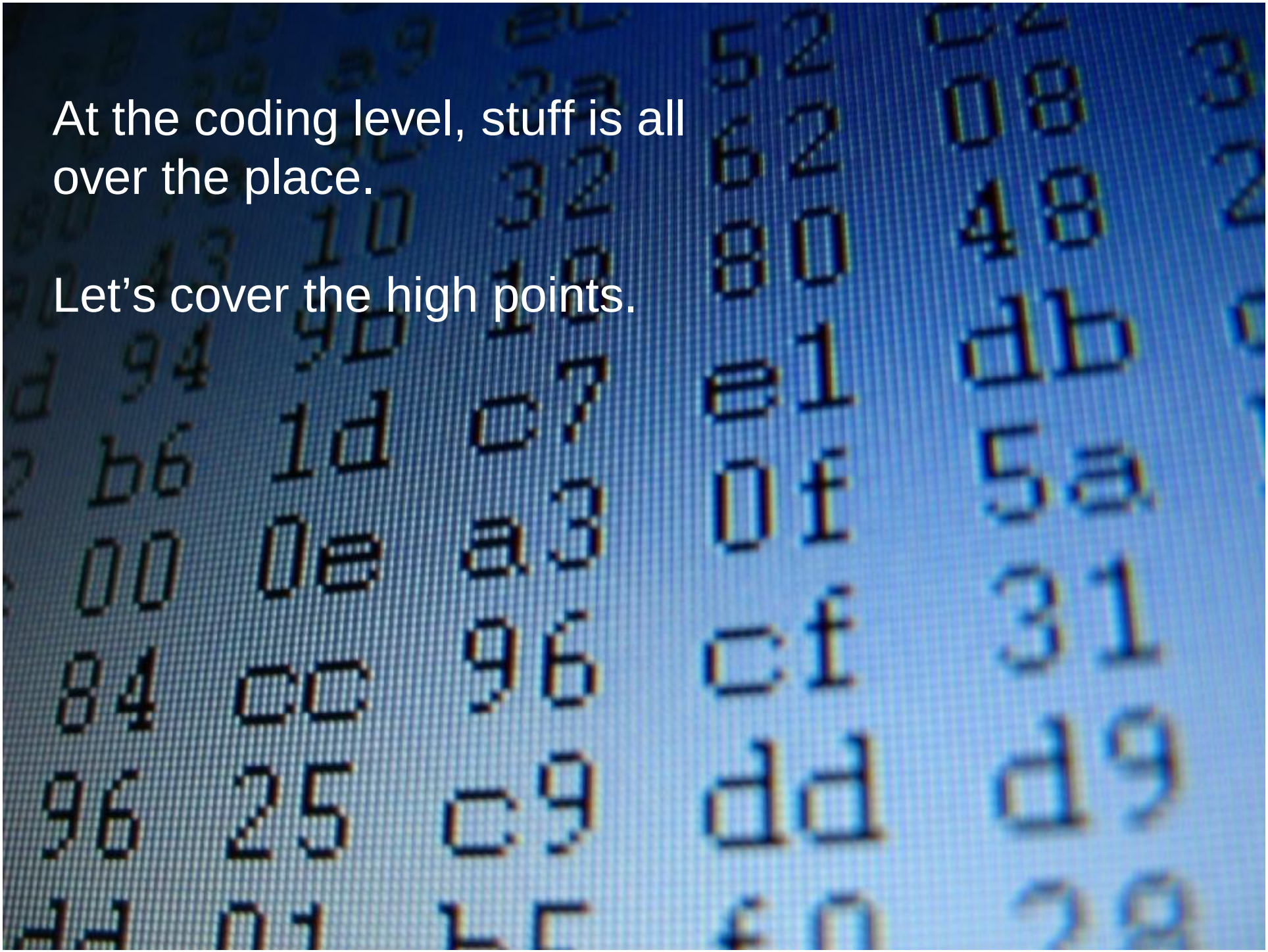
… There's almost no hurdle to jump through on Mac OS X."

- Charlie Miller, 2009 CANSECWEST Pwn2Own contest winner, enjoying his new Mac Book Pro.

© penny-arcade.com **J**

Conceptually, think of metasploit exploitation in three primary sections.

- Exploits
  - gain control of EIP
- Payloads
  - force EIP to do your evil bidding
- Everything else (auxiliary, dos, the dev stuff, etc)
  - these are the scanners, uber modules that combine sub-exploits, dev tools, etc


At the coding level, stuff is all over the place.
Let’s cover the high points.

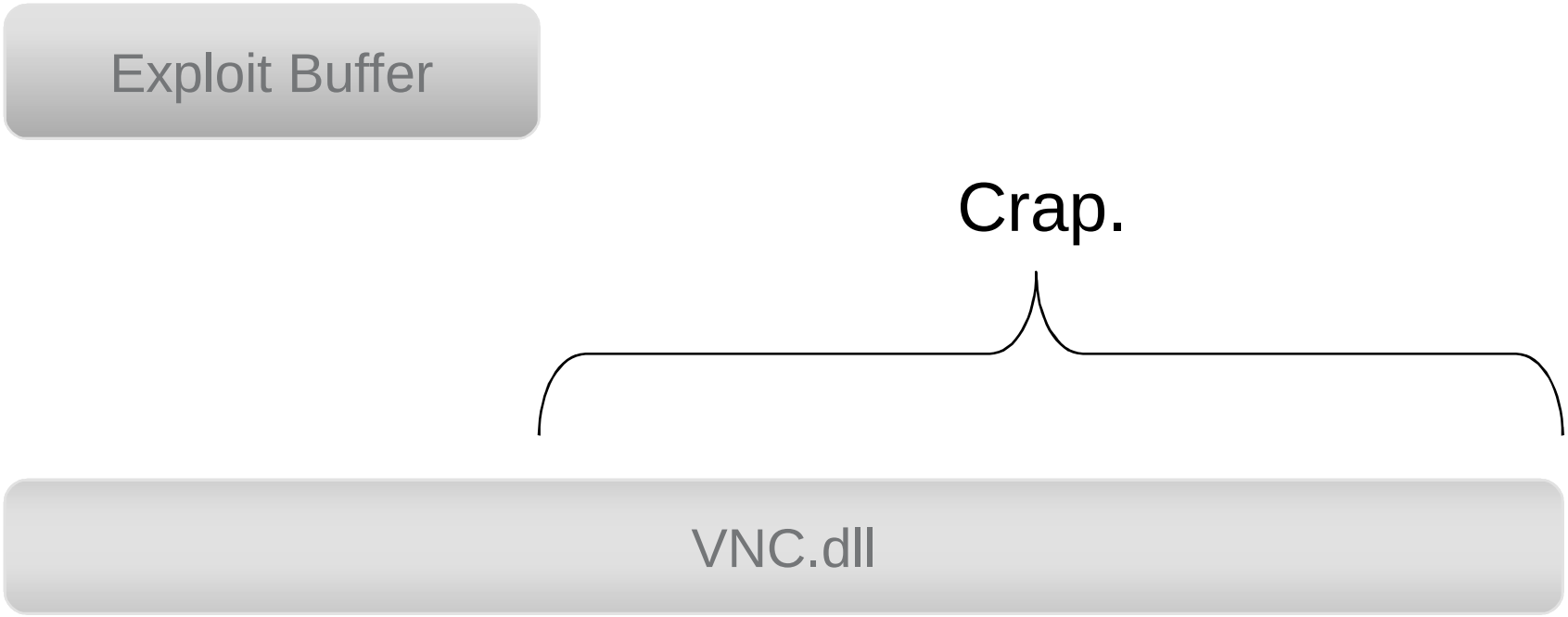
These aren’t all the code locations, but this is what you need to know for today.

exploits/	encoders, exploits, payloads, and auxiliary mods go here
external/	'external' source code: meterpreter VC++, passivex VC++, ASM shellcode
data/	compiled or unmodified stuff that gets pushed down (meterpreter.dll, etc)
documentation/	HAHAHAHAHA, oh, that's good.
lib/msf/core/handler/	handling functions for payloads and stages

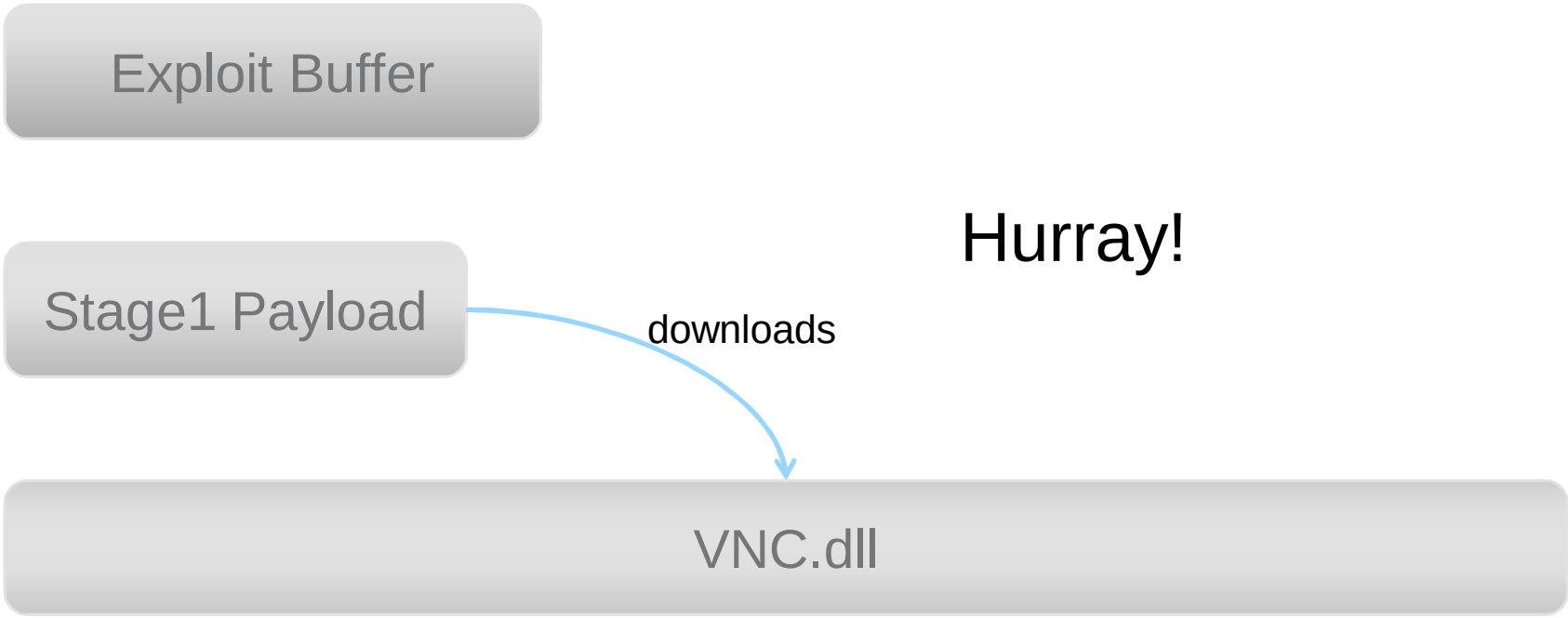
Payloads are split up into ‘stages’ to get around technical restrictions

- Stagers allow you to use arbitrary size payloads with limited size exploits
- Almost all exploits have tight size restrictions for the initial payload
- MSF gets around this by pushing down initial stagers with size < ~300 bytes, ideally < 100 bytes (this is from memory, don’t shoot me if I’m off)

A VNC payload would never fit on it’s own; it’s way too big.



A VNC payload would never fit on it’s own; it’s way too big.



Payloads come in a variety of fun connecting flavors, too

```
Windows Meterpreter, Bind TCP Stager (IPv6)
Windows Meterpreter, Bind TCP Stager (No NX Support)
Windows Meterpreter, Bind TCP Stager
Windows Meterpreter, Find Tag Ordinal Stager
Windows Meterpreter, PassiveX Reverse HTTP Tunneling Stager
Windows Meterpreter, Reverse TCP Stager (IPv6)
Windows Meterpreter, Reverse TCP Stager (No NX Support)
Windows Meterpreter, Reverse Ordinal TCP Stager
Windows Meterpreter, Reverse TCP Stager
Windows Meterpreter, Bind TCP Stager (IPv6)
Windows Meterpreter, Bind TCP Stager (No NX Support)
Windows Meterpreter, Bind TCP Stager
Windows Meterpreter, Find Tag Ordinal Stager
Windows Meterpreter, PassiveX Reverse HTTP Tunneling Stager
Windows Meterpreter, Reverse TCP Stager (IPv6)
Windows Meterpreter, Reverse TCP Stager (No NX Support)
Windows Meterpreter, Reverse Ordinal TCP Stager
Windows Meterpreter, Reverse TCP Stager
```

Lesson one in descriptive naming schemes: bind_tcp payloads bind a tcp port.

- Execute the payload and map it’s IO to a bound port
- Similar to ‘nc -l -p 31337 -e cmd.exe’
- Useful for demos, internal environments, or on compromises of externally facing servers

```
windows/meterpreter/bind_ipv6_tcp              Windows Meterpreter, Bind TCP Stager (IPv6)
windows/meterpreter/bind_nonx_tcp              Windows Meterpreter, Bind TCP Stager (No NX Support)
windows/meterpreter/bind_tcp                   Windows Meterpreter, Bind TCP Stager
```

Lesson two in descriptive naming schemes: reverse_tcp payloads spawn a reverse tcp connection.

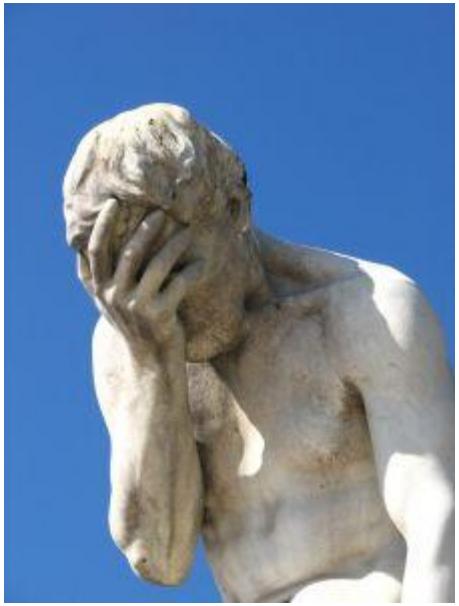
- In most corporate (and home) environments, firewalls do not filter outbound connections on common ports, such as 20/21, 22, 53, 80, 443.

- A reverse_tcp payload takes advantage this by reaching across the Internet to your msf server, side stepping the firewall / NAT router on the perimeter

```
windows/reflectivemeterpreter/reverse_ipv6_tcp  Windows Meterpreter, Reverse TCP Stager (IPv6)
windows/reflectivemeterpreter/reverse_nonx_tcp  Windows Meterpreter, Reverse TCP Stager (No NX Support)
windows/reflectivemeterpreter/reverse_ord_tcp   Windows Meterpreter, Reverse Ordinal TCP Stager
windows/reflectivemeterpreter/reverse_tcp       Windows Meterpreter, Reverse TCP Stager
```

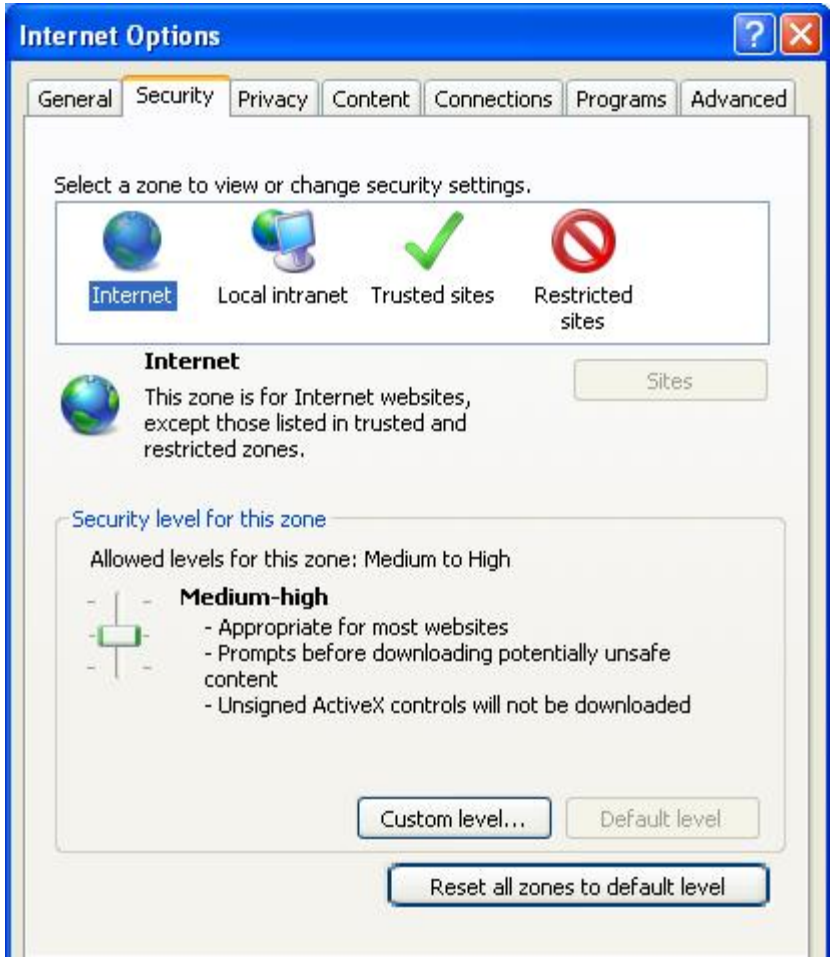
What other scenarios do we commonly see in enterprise environments?

- How about when the corporate network does proper egress filtering?

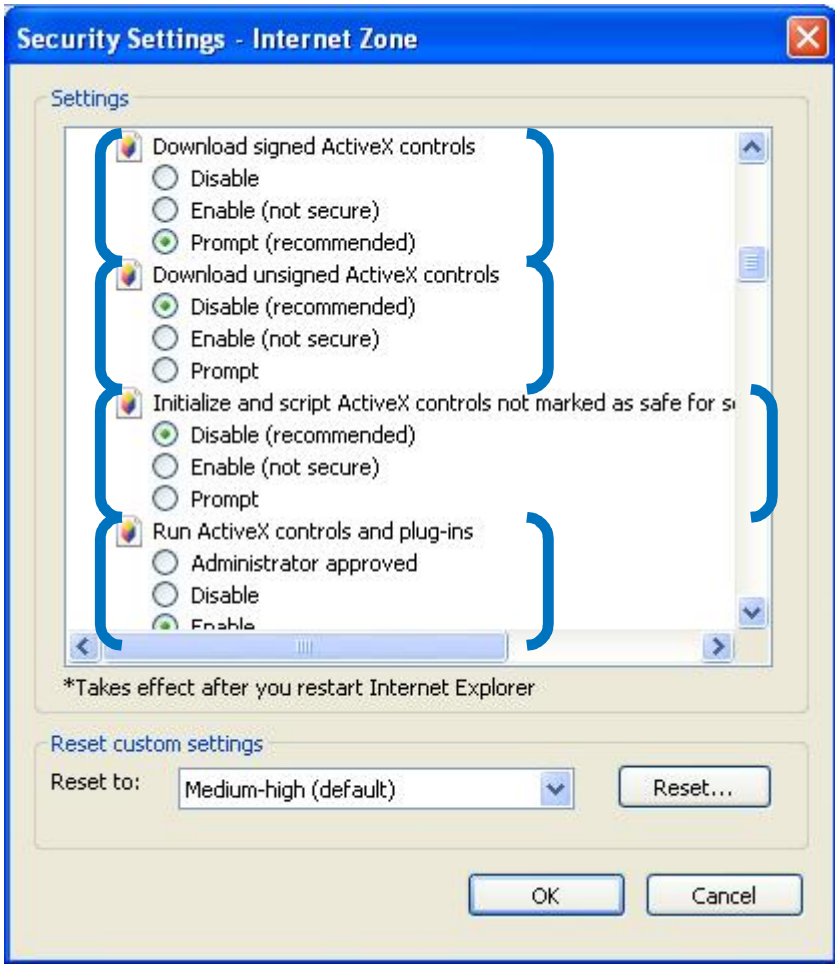


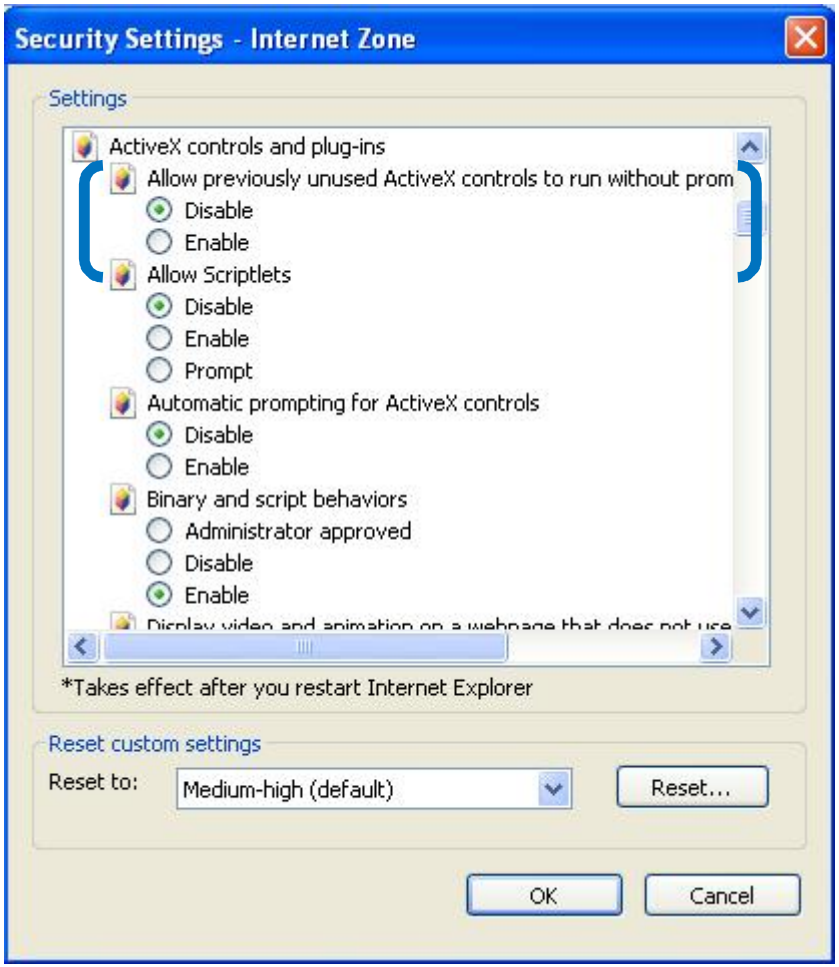
We’ll get there, but let’s detour into the underbelly of the beast for a slide or two.

Internet Explorer’s security architecture uses zones at it’s heart.



Along with zones, IE uses magic juju to control access to activex controls.

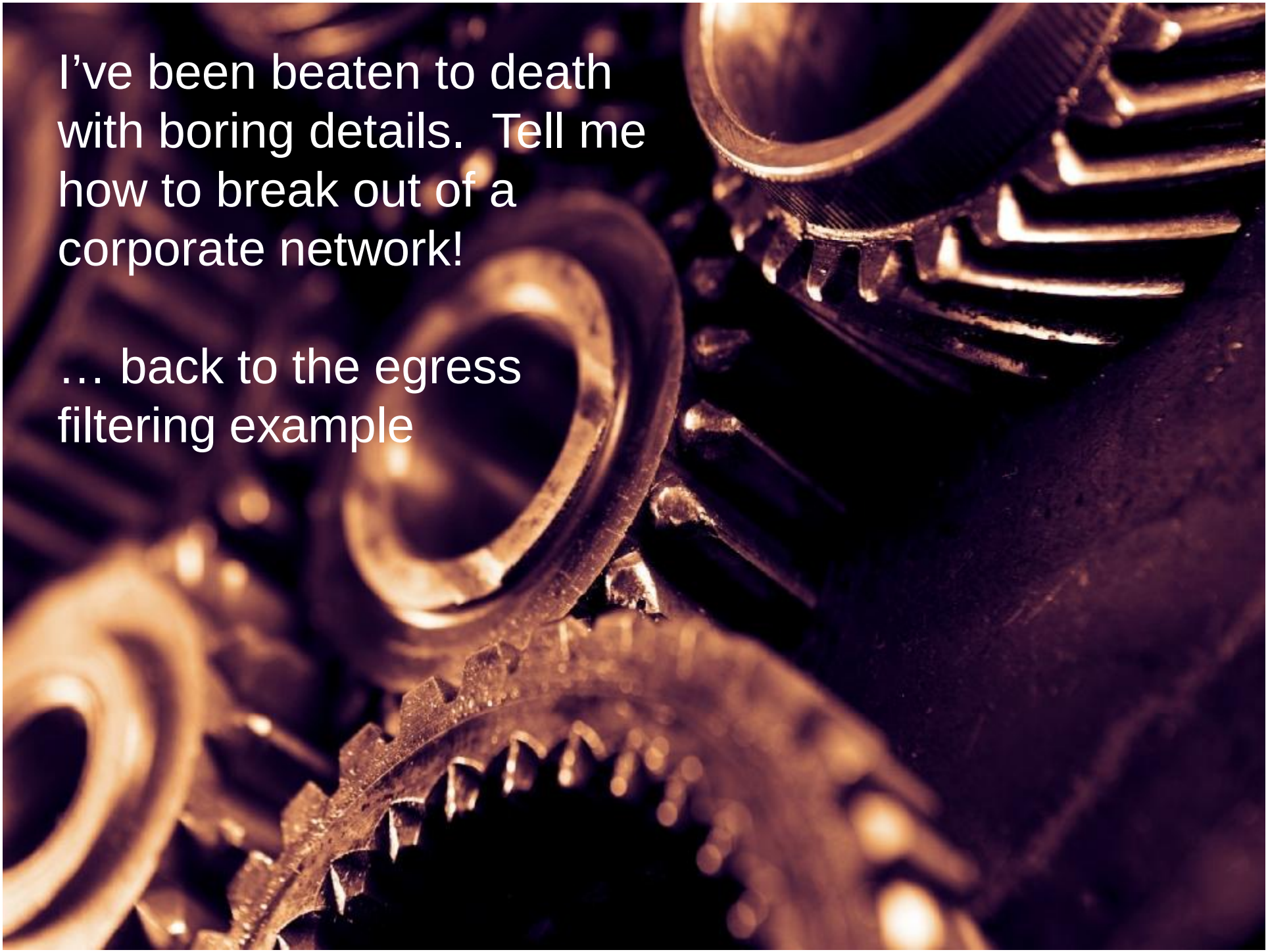




Along with zones, IE uses magic juju to control access to activex controls.

HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\{0,1,2,3,4}

- 1001, Download signed ActiveX controls
- 1004, Download unsigned ActiveX controls
- 1200, Run ActiveX controls and plug-ins
- 1201, Initialize and script ActiveX controls not marked as safe for scripting
- 1208, Allow previously unused ActiveX controls to run without prompt (disabled in IE7)


I’ve been beaten to death with boring details. Tell me how to break out of a corporate network!
… back to the egress filtering example

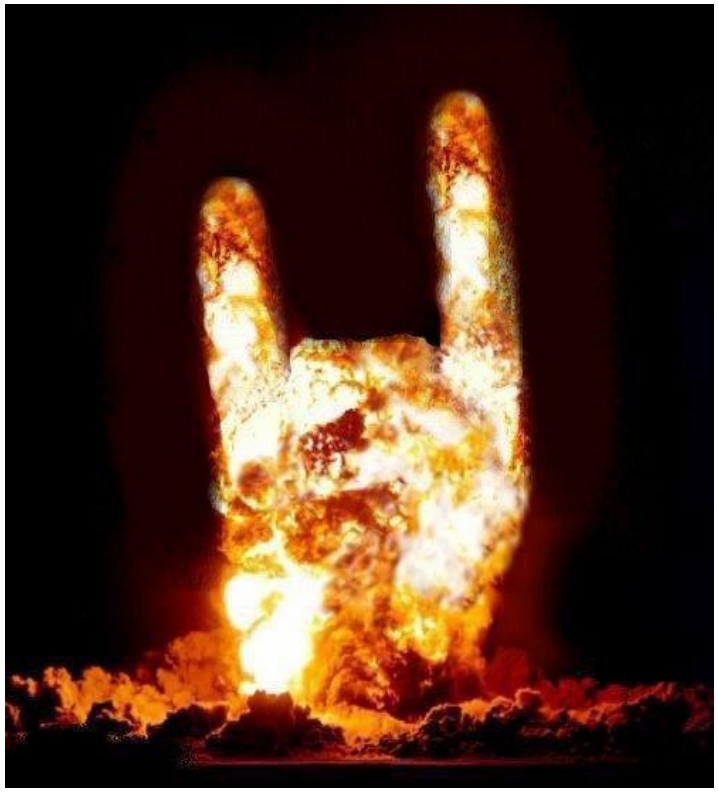
Raise your hand if you're here because you want to know how to pwn corporate networks. K, the police are here to escort you away.

- If your browser can talk to the Internet then there's an active comm channel available, even if it's not obvious.

- skape (http://www.hick.org/~mmiller/) implemented an msf stager / comm channel that runs inside hidden copies of Internet Explorer; he called it 'passivex'

- Excellent write-up: "Post-Exploitation on Windows using ActiveX Controls" (http://uninformed.org/index.cgi?v=1&a=3)

But this is DC405 – how's this thing work?



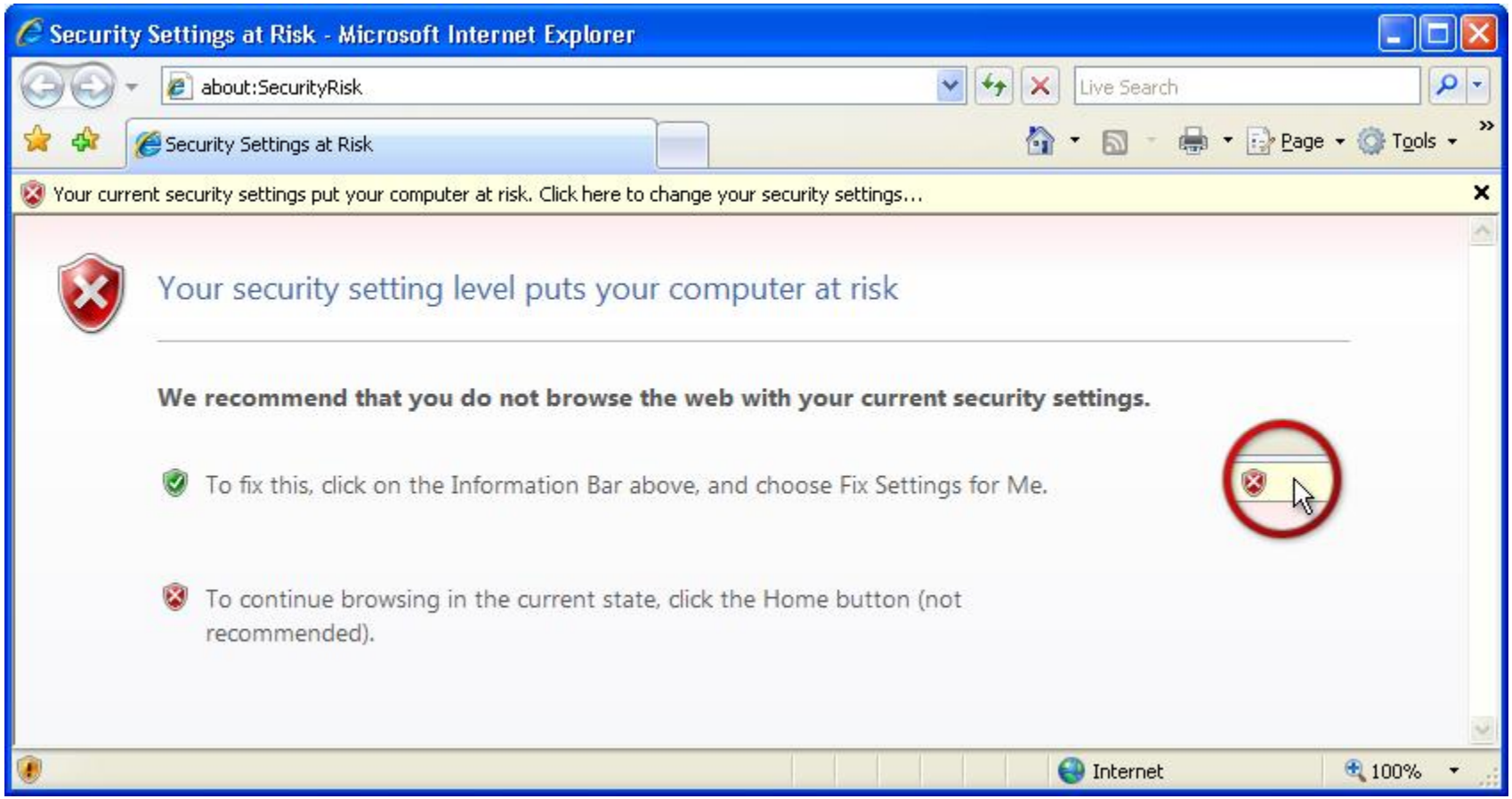
PassiveX pre-2009:

- initial stager modifies INTERNET zone security settings, launches iexplore.exe pointed towards MSF server
- msf pushes down the activex control (passivex.dll)
- passivex takes over, creates a comm channel through IE back to MSF, pulls down target payload
- transfers control to target payload

Unfortunately, all was not good throughout the land.

- The initial stager modified 4 registry settings needed for IE6 (1001,1004,1200,1201)
- IE7 introduced a changed default security setting that broke passivex (1208)
- Initial stager was almost the maximum size it could be, so devs were reluctant to add the 5th setting. (Which is good, because my ASM is atrocious.)
- So passivex sat broken (for IE7, the most popular browser) and unused for a long time.

Btw, not only did it not work with IE7+, but it severely gimped the INTERNET security zone, guaranteeing random pwnage. (Not good.)




Don't worry
We can take over the world soon enough

The new IE7 setting blocks us from loading *new* activex controls.

- 1208, “ActiveX controls and plug-ins: Allow previously unused ActiveX controls to run without prompt” became disabled by default in IE7.

- What if we don’t have to load a new control? What if whatever we want to use is already there; maybe part of the OS?

So I started looking at other ways to get this working with IE7.

Remind me, what permissions do we have at this point?

Setting Value Name	Description
1001	Download signed ActiveX controls
1004	Download unsigned ActiveX controls
1200	Run ActiveX controls and plugins
1201	Initialize and script ActiveX controls not marked as safe for scripting

(from the uninformed article)

So I started looking at other ways to get this working with IE7.

Remind me, what permissions do we have at this point?

Setting Value Name	Description
1001	Download signed ActiveX controls
1004	Download unsigned ActiveX controls
1200	Run ActiveX controls and plugins
1201	Initialize and script ActiveX controls not marked as safe for scripting

(from the uninformed article)

I recognized 1201 -- this thing grants access to cool system functions that are used all over the place in vbscript

More specifically:

- WScript.Shell
  - Execute OS commands, access the registry
- WScript.FileSystemObject
  - Read/Write/Modify text files
- And actually, I had already written an exploit for that setting (windows/browser/ie_unsafe_scripting)

This means IE already had access to do everything it needed.

- It just needed to be told to use it.
- Whipped up the new javascript handling routine and implemented in lib/msf/core/handler/passivex.rb
- Pushed to svn 3 weeks ago (yay!)

Specifically, the javascript temporarily lowers IE's security for our IP address, then puts everything back the way it found it.

1. Maps our IP address to the INTRANET zone
2. Modifies 5 perms for the INTRANET zone (1001, 1004, 1200, 1201, 1208)
3. Cleans up after the initial stager, fixing the INTERNET zone
4. Launches another copy of iexplore.exe pointed at msf
5. Waits 60 seconds, deletes INTRANET settings

Walk through code from:
lib/msf/core/handler/passivex.rb
… then DEMO!

And if we have time, walk
through ie_unsafe_scripting.rb

Where to next?

- Clean up code and make it less signature prone.
- IE8’s giving me headaches. I’m not positive, but I think it’s running my code inside a sandbox of some sorts. (My IE8 install may just be borked.)
- I’m considering trying to implement passivex as a regular com object outside of IE to quit getting nailed by IE changes. 1201 would be all that was needed.

Questions? Complaints?

email: natron [at] metasploit.com
jabber / gtalk / etc: natron [at] invisibledenizen.org

http://blog.invisibledenizen.org

Give me some time and I'll post these slides
up on my blog and the dc405 forum.