That MITM Talk with the Disgusting Title

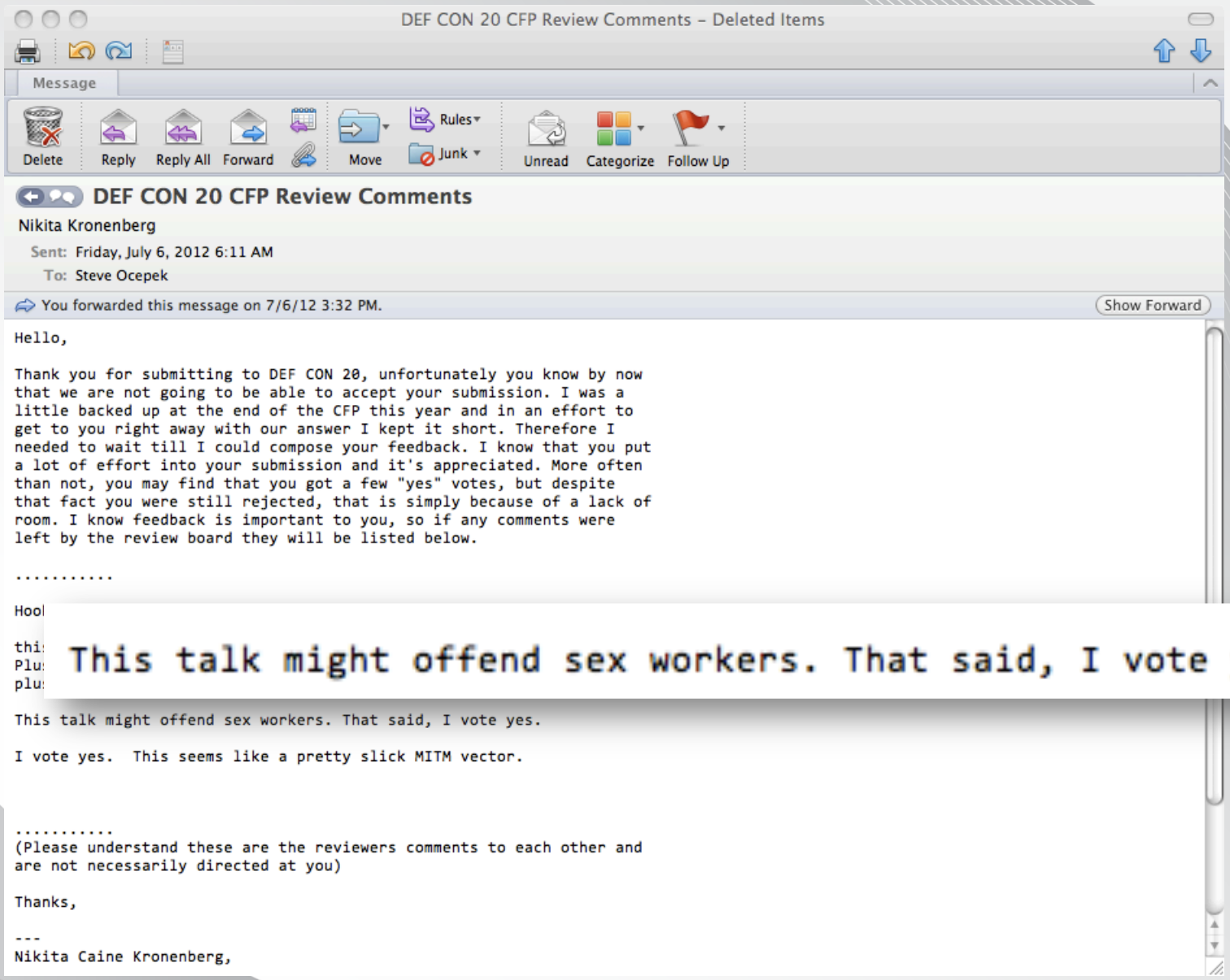
Presented by:

Ryan Linn
Steve Ocepek
Trustwave SpiderLabs

CONGRATULATIONS

Trustwave SpiderLabs

Trustwave



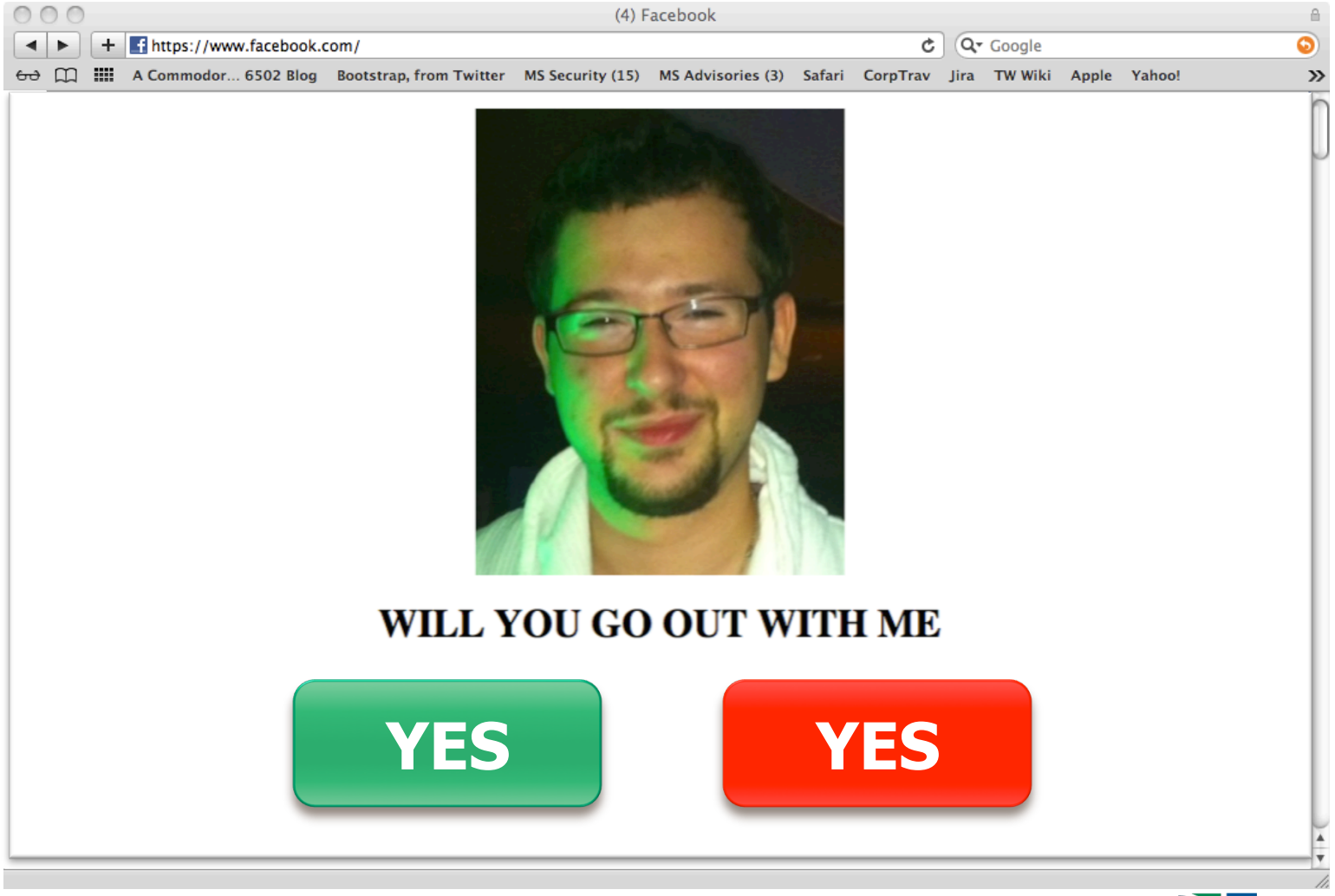
The Story So Far

- Boy meets girl
- Boy tries to impress girl
- Hilarity
- Girl goes out with foreign exchange student
- 80's music video montage

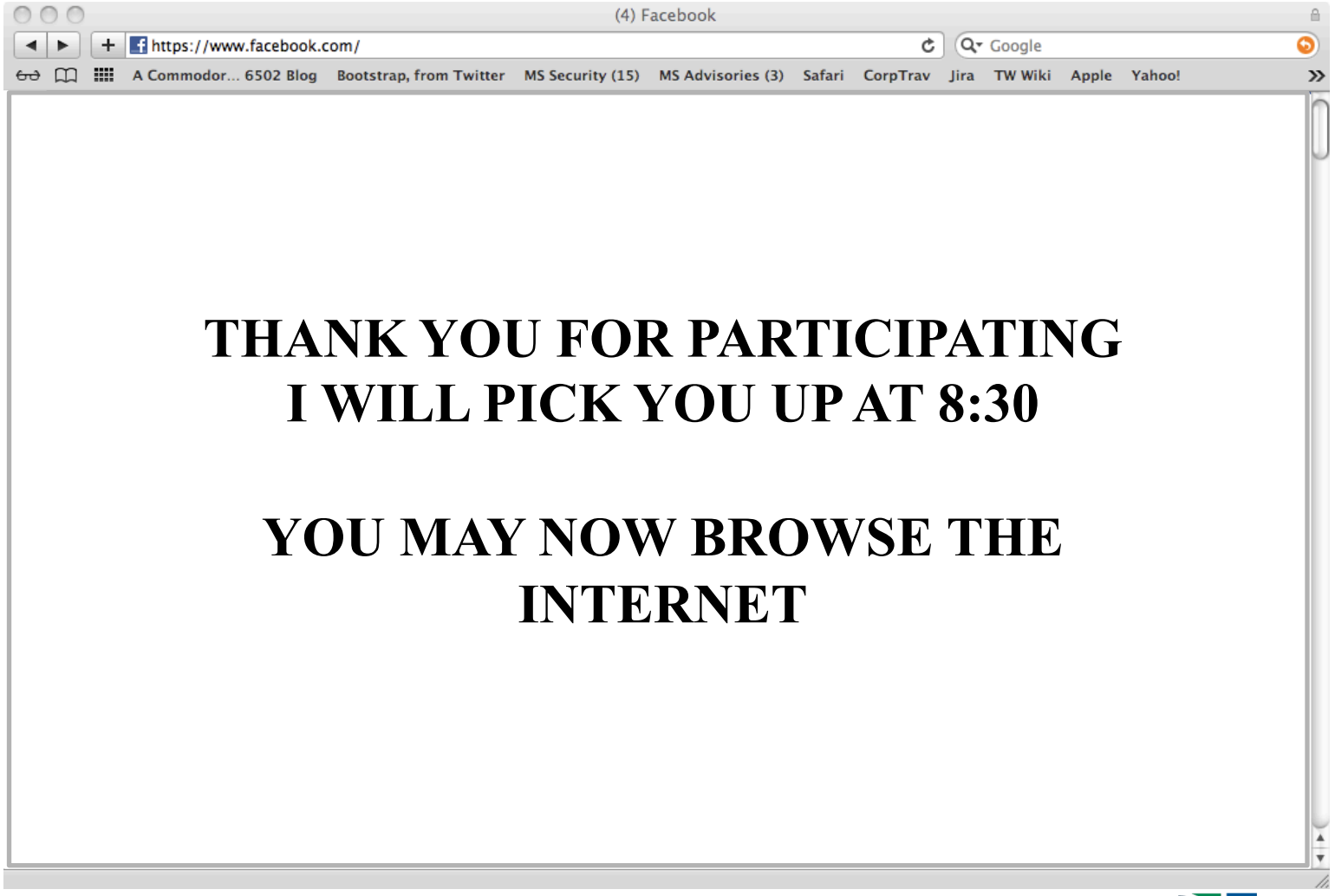
Boy Becomes a Man, in the Middle

- Boy installs BT5
- Boy follows girl into coffee shop
- Boy uses ettercap to inject traffic into HTTP stream

An Offer She Cannot Refuse



An Offer She Cannot Refuse



Trustwave SpiderLabs

Trustwave®

How a Boy Becomes a Man, In the Middle

Trustwave SpiderLabs

Trustwave

Girl

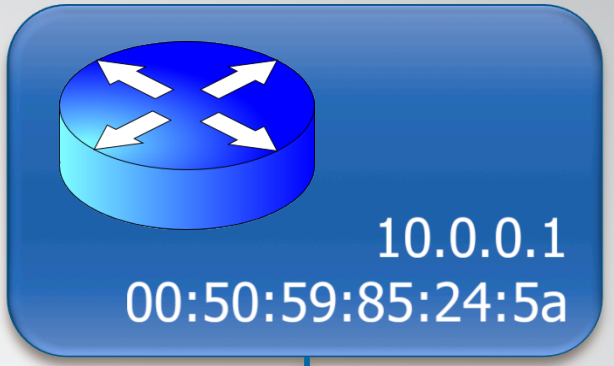
How a ~~Boy~~ Becomes a Man, In the Middle

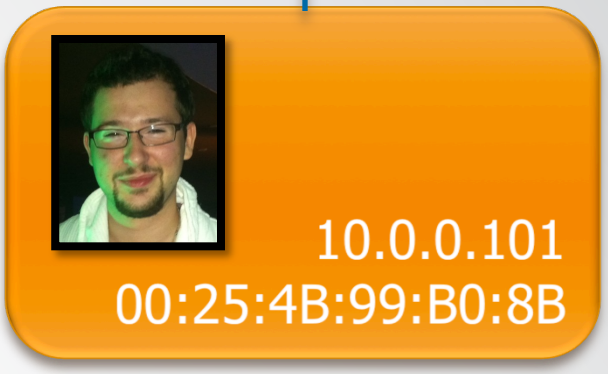
Trustwave SpiderLabs

Trustwave®

Girl Madam

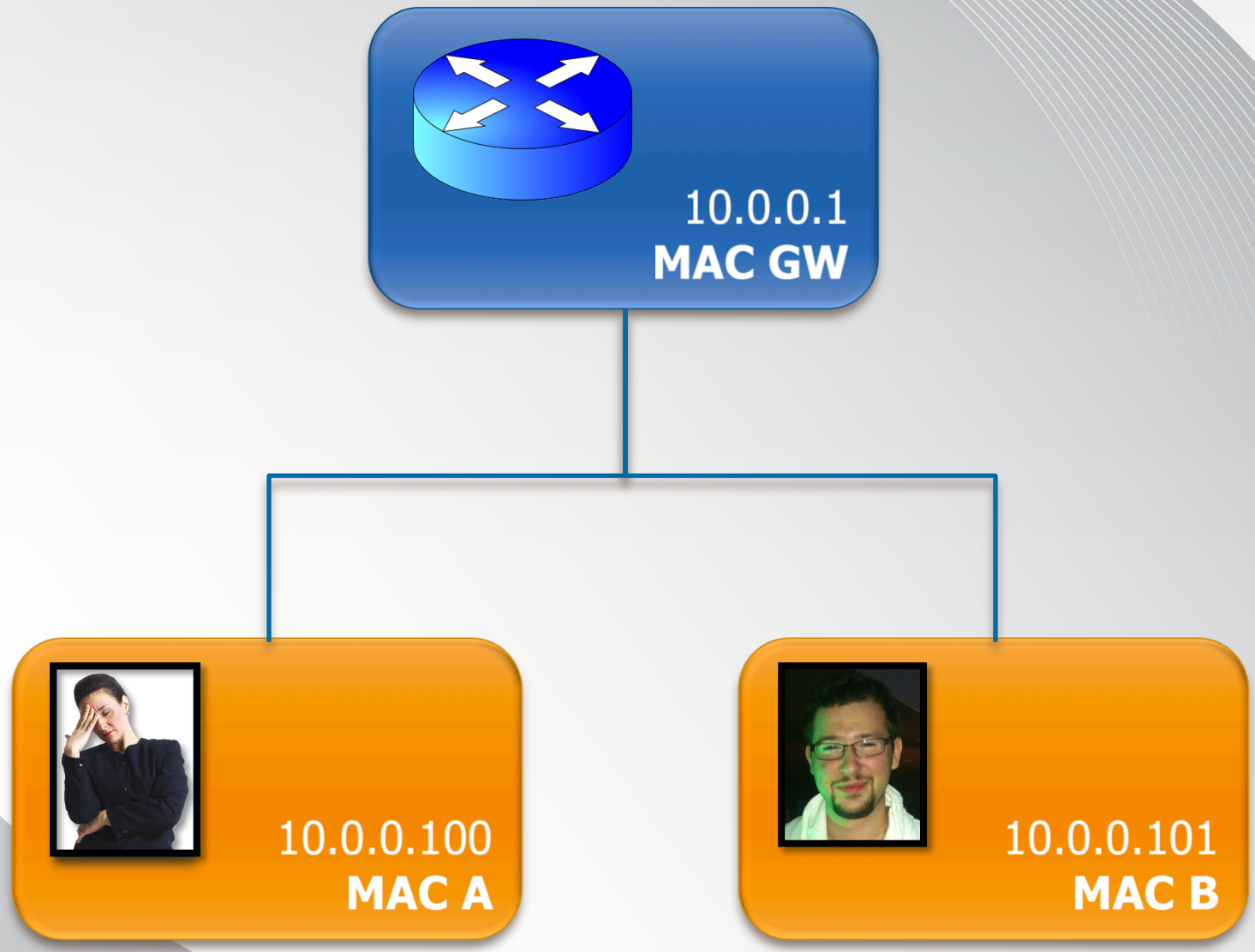
How a ~~Boy~~ Becomes a ~~Man~~, In the Middle





Trustwave SpiderLabs

Trustwave®

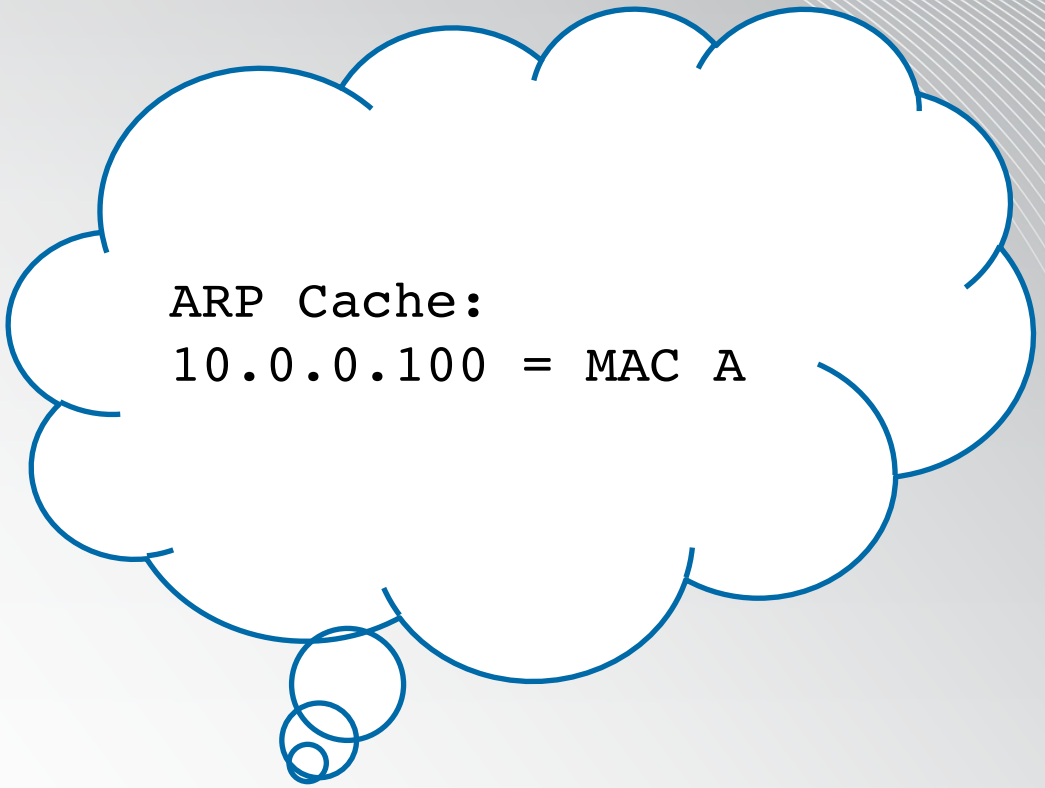


Trustwave SpiderLabs

Trustwave®

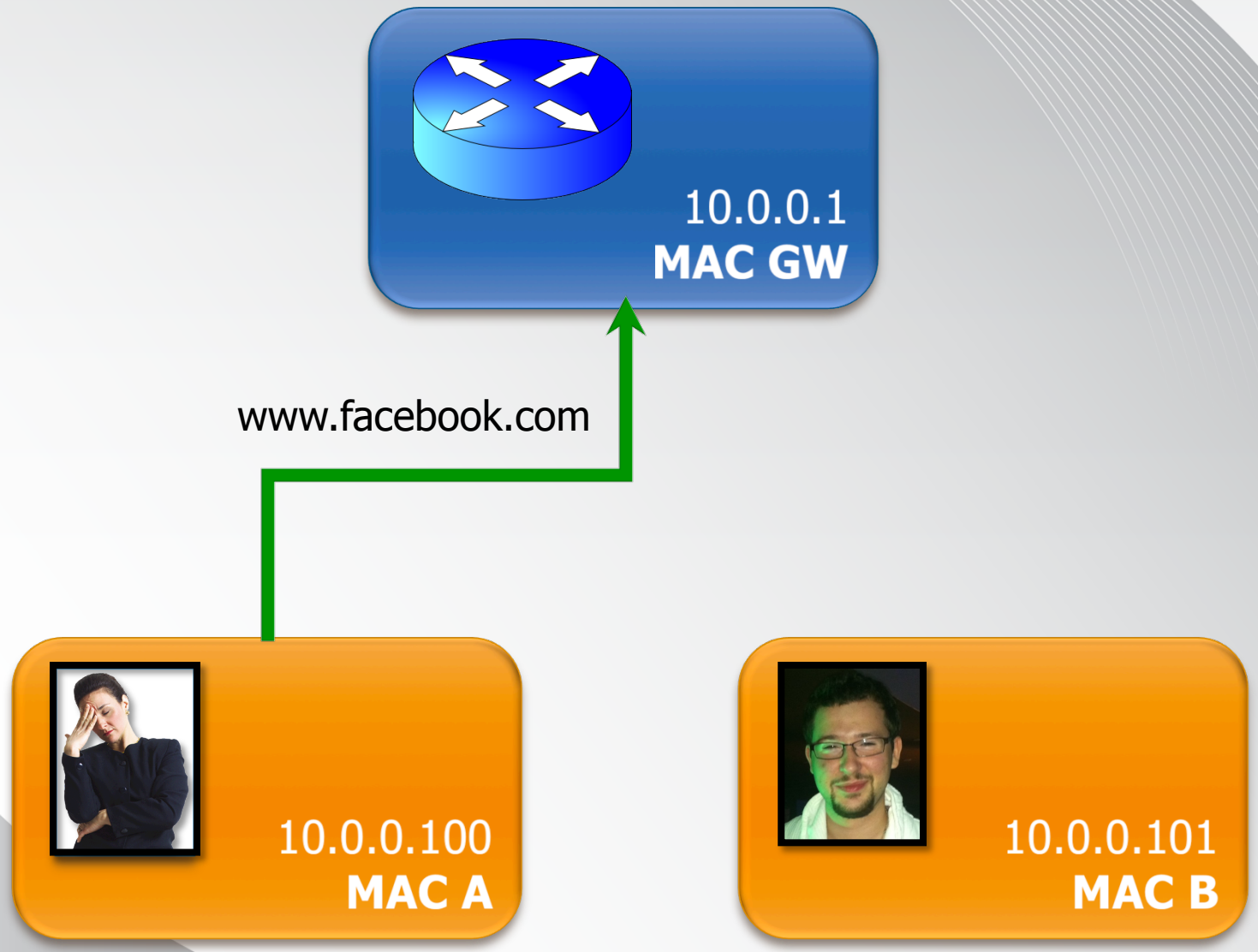
ARP Cache:
10.0.0.1 = **MAC GW**

10.0.0.100
MAC A



Trustwave SpiderLabs

Trustwave®



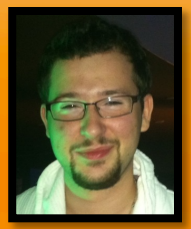
Trustwave SpiderLabs

Trustwave®

ARP Request:

Who has 10.0.0.100?
Tell **10.0.0.1** at **MAC B**

10.0.0.100
MAC A



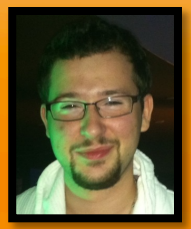
10.0.0.101
MAC B

Trustwave SpiderLabs

Trustwave

ARP Cache:
10.0.0.1 = **MAC B**

10.0.0.100
MAC A



10.0.0.101
MAC B

Trustwave SpiderLabs

Trustwave®

www.facebook.com

Trustwave SpiderLabs

Trustwave®

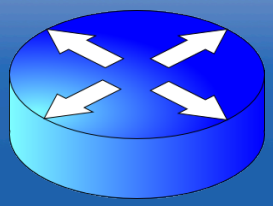
ARP Request:

Who has 10.0.0.1?
Tell **10.0.0.100** at **MAC B**

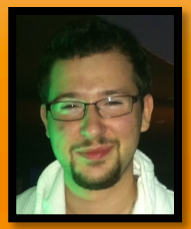
Trustwave SpiderLabs

Trustwave®

ARP Cache:
10.0.0.100 = **MAC B**



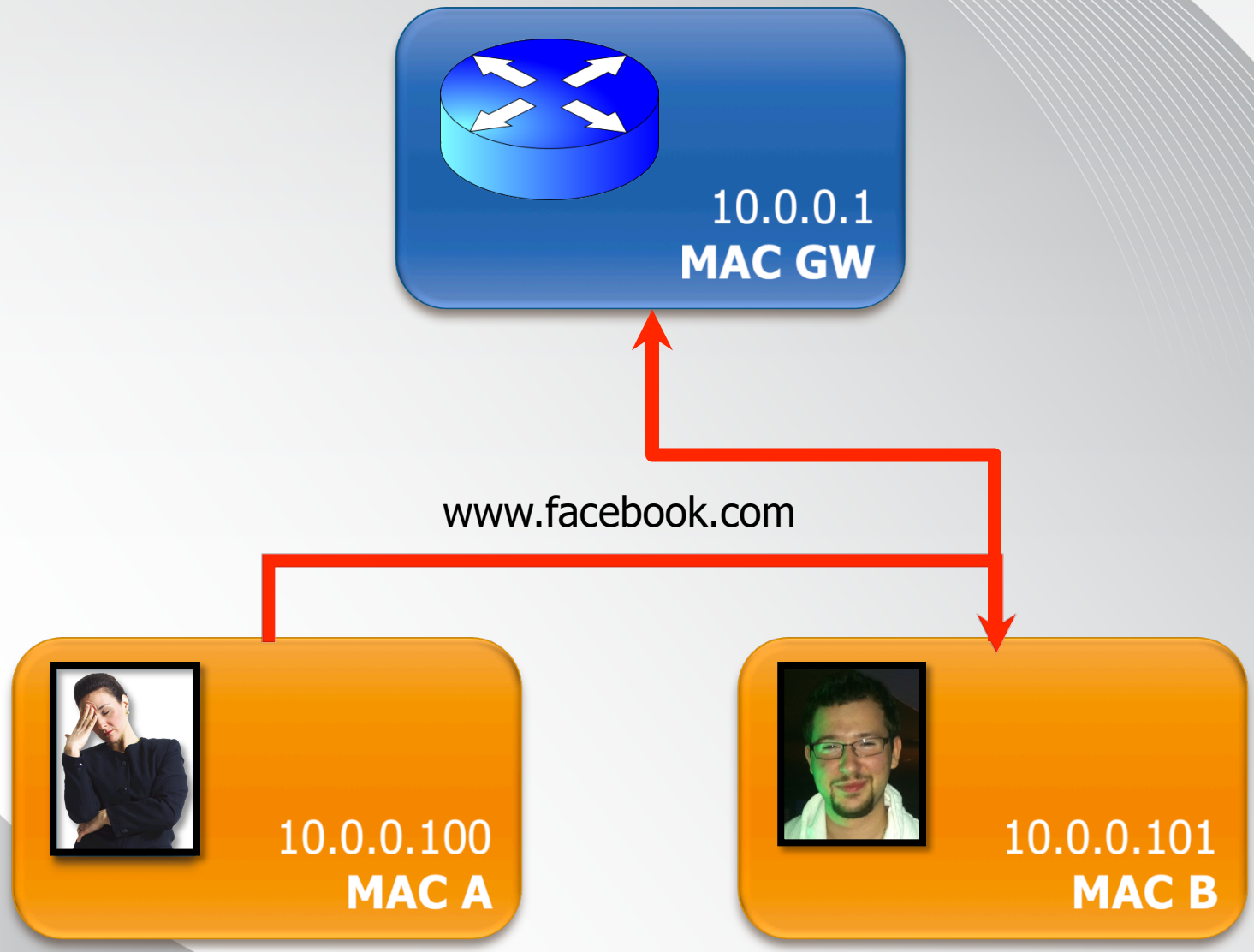
10.0.0.1
MAC GW



10.0.0.101
MAC B

Trustwave SpiderLabs

Trustwave



Trustwave SpiderLabs

Trustwave®

But now what?

- Inject traffic into stream
  - UNC Path injection
  - Redirect to active site - session cookie stealing
- Or just be a creep and watch

Why Does it Still Work?

- Infrastructure problems
- Security measures are available, but rarely employed
- Often considered to be unpreventable "insider threat", like sitting at keyboard of server
- Girls need to get to Facebook, always click YES eventually

Real World

- Great way to pivot once access is obtained
- Indications during investigations that these methods are used
  - Proof is hard to come by, because ARP is rarely logged
- Commonly used in pentests
  - Great for recon data
  - The notorious UNC injection

ettercap

- Feature-rich MITM framework
- etterfilter allows modification on the fly
- Recently picked up by new maintainers
- Created by ALoR and NaGa

ALoR and NaGA

SSLStrip

- Moxie's tool for changing https: references to http:
- He recommends arpspoof for the actual MITM, arpspoof does the rest

thicknet

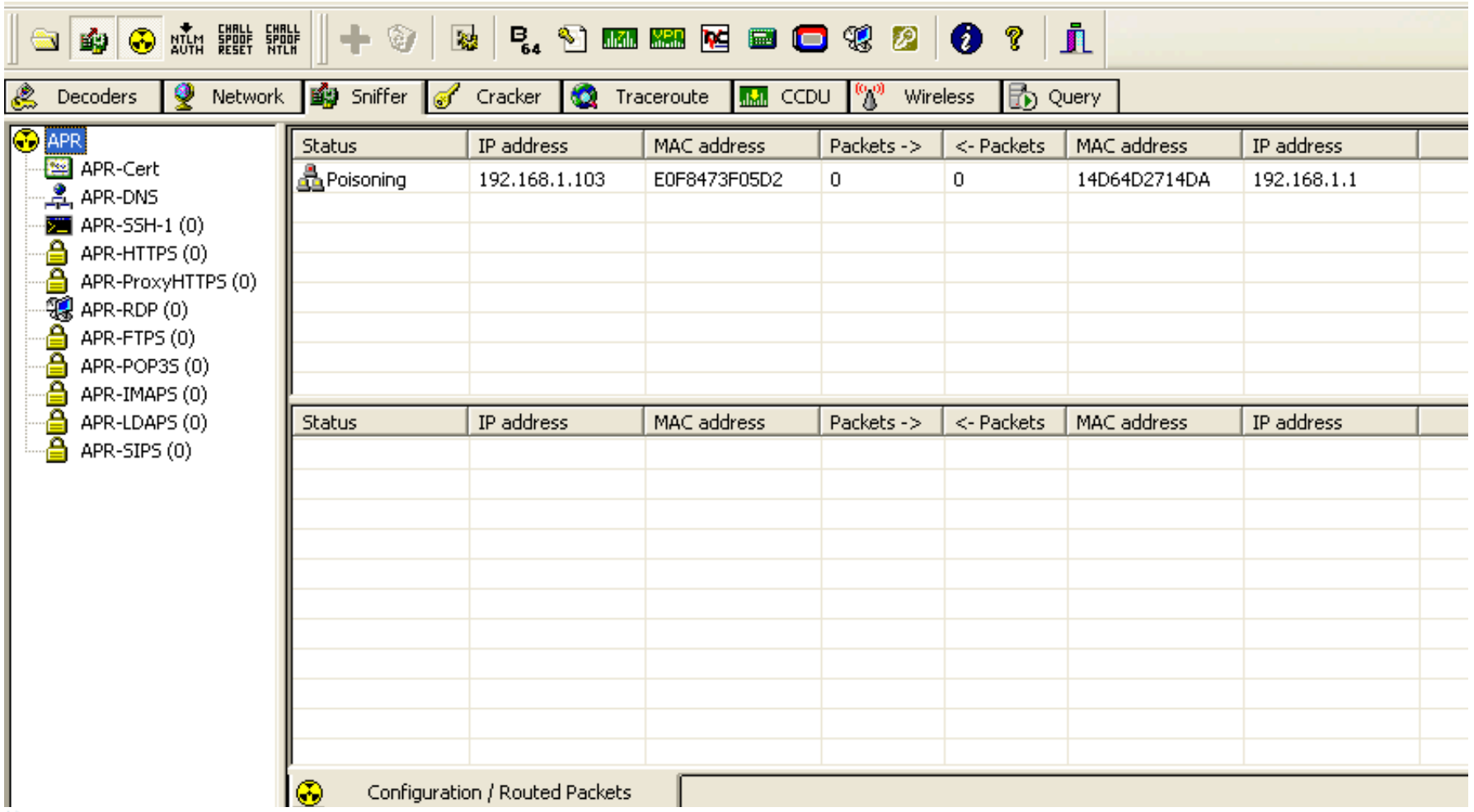
- First MITM session takeover framework
- Wendel Henrique and I created it for Oracle and MSSQL engagements
- Currently in licensing talks for a new line of men's body spray

** Picture by Felix Jackson, Jr.*

Cain and Abel

- Windows based tool for sniffing, MITM, and credential cracking
- Supports dozens of different types of hashes
- Calculators for RSA tokens and Cisco type 7 hashes
- Point and click MITM

Cain and Abel



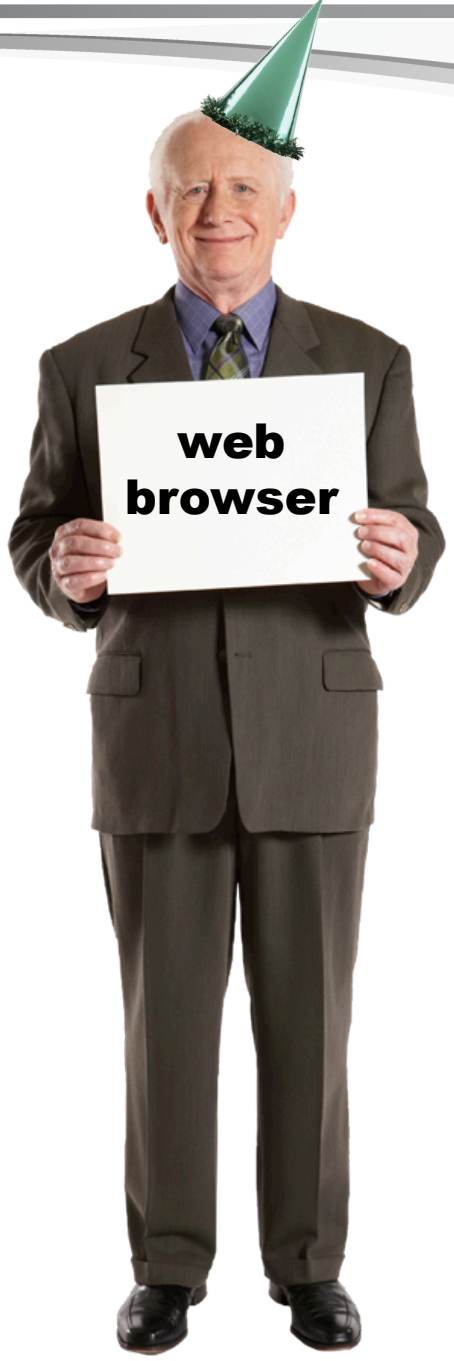
BeEF and Owning Browsers

Any Ideas?

- If we were to target a network client application, which one would it be?
- Preferably one with the ability to execute inline code.
- Hmmmmm….
- Any ideas???

Trustwave SpiderLabs

Trustwave®



Trustwave SpiderLabs

Trustwave®

Focus is good

- thicknet is hard, every protocol needs its own implementation
- The whole web model thing lets us put code into packets and execute it
  - Javascript, Flash
- Would be nice if there was a project that made this easy

Yay BeEF!

Browser Exploitation Framework

"BeEF will hook one or more web browsers and use them as beachheads for launching directed command modules and further attacks against the system from within the browser context."

What is BeEF?

- Allows an attacker to control browsers
- Attacker "hooks" browsers, then called "zombies"
- Interface with the browser is in JavaScript
- Module design to easily extend new functionality
- Interface with Metasploit
- Cross-Protocol Exploits
- Proxy
- And more…

BeEF Injection? Surprise!

Trustwave SpiderLabs

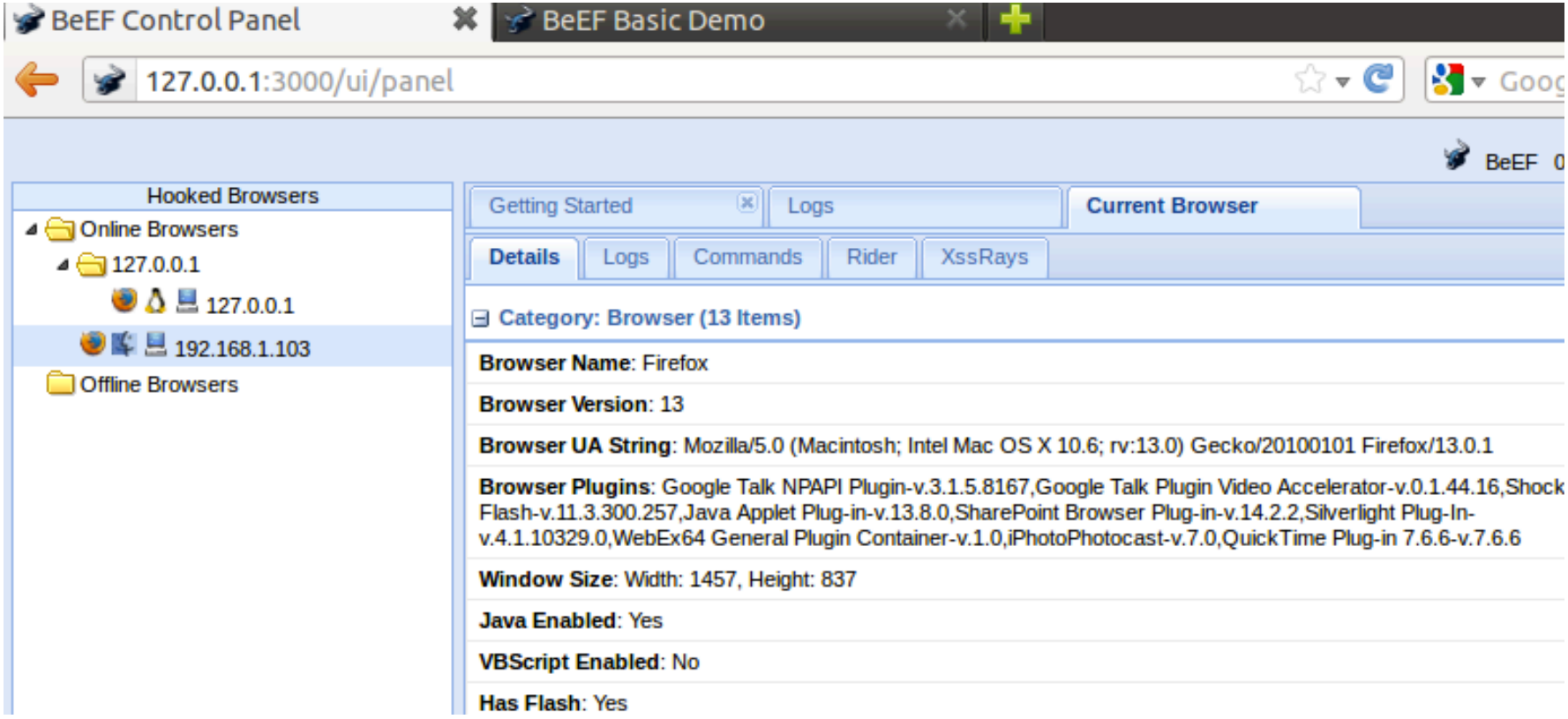
* http://www.public-domain-image.com/animals/cow-buffalo/slides/red-and-white-cow-on-green-grass.html

Trustwave®

BeEF Injection

- Enticing a browser to run JavaScript code
- Beacons back to BeEF server
- When new actions are available to run, executes code, response back with results
- Typical injection points
  - Persistent XSS
  - XSS + Phishing
  - Phishing + malicious site

BeEF Console



Trustwave SpiderLabs

Trustwave®

BeEF Challenges

- You can't put it just anywhere
- You have to find vulnerable web pages
- Or make them
- Have to get people to browse to your links
- Phishing may tip off that there's a problem
- Hooking is typically transient, your zombies may shamble away

BeEF Injection using Man In The Middle

(picture redacted)

ettercap is awesome, but…

- Unique issues with injecting web content
- Limited to search and replace **without changing packet size**
- Disable gzip hacks
  - Accept-Nanerpus

The Nanerpus

```
if (tcp.dst == 80) {
    if (replace("Accept-Encoding", "Accept-Nanerpus"))
}
```

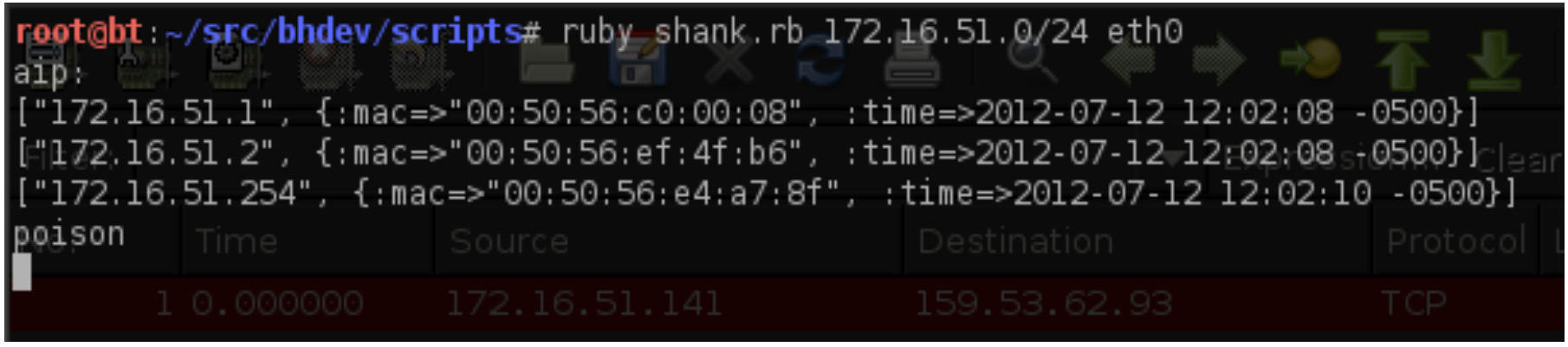
* http://www.flickr.com/photos/red5standingby/167042582/

thicknet is awesome, but…

- Made for interactive session injection
- Focused on DB protocols currently
- Maybe a bit overkill for this use case
- It's in Perl
  - Ruby wahhhhh

FINE I WILL MAKE YOU A TOOL AND IT WILL BE IN RUBY

Let's call it shank



Trustwave SpiderLabs

Trustwave®

Features

- Easy ARP poisoner / forwarder
- Stateful
- Few external dependencies
  - PacketFu
- It sets Accept-Encoding: identity
  - No more nanerpus!
  - (Actual gzip mod is hard,hard,hard :P)

- shank is also a great verb

Ruby + MITM ~~Performance~~

- This was actually kind of a paint in the ass
- Heavy use of bpf filters
  - All your favorites like
  - `(tcp[(tcp[12]>>2):4] = 0x47455420 or tcp [(tcp[12]>>2):4] = 0x48545450) and port 80`
  - GET or HTTP matching, in bpf!
- Multiple pcap sessions, multiple bpf filters
  - Let pcap do the work
  - Packet.parse() makes fan go WIRRRRRRR

BeEF Integration

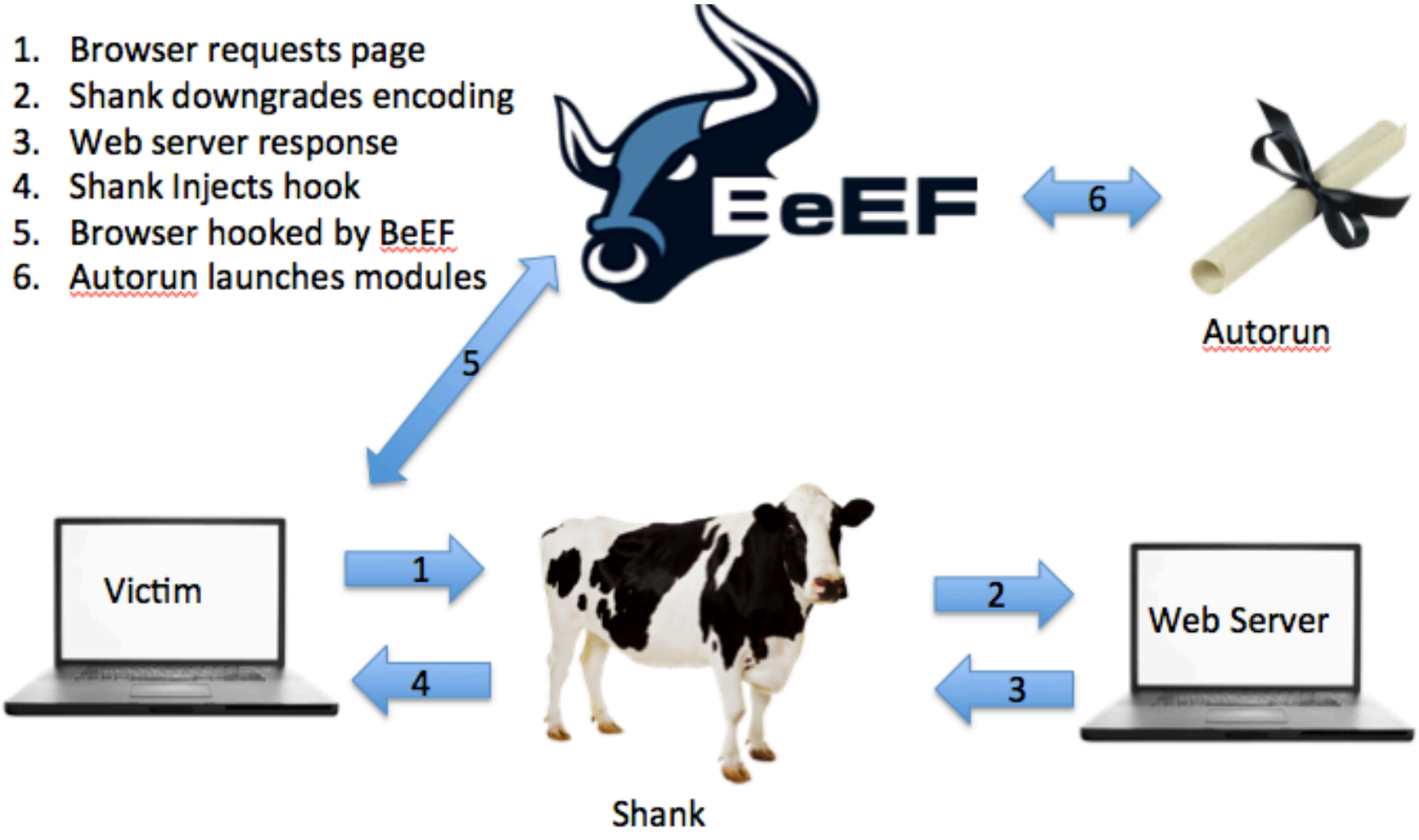
- Through BeEF's rest API, we can identify hooked browsers
- Only poison hosts that aren't active zombies
- Limit impact on the browser
- Hook anything talking HTTP that can do JavaScript
- Maintain hooks even if people leave pages

BeEF + Autorun

- When browsers become zombies, a module can be run against them automatically
- Autorun allows multiple modules to be run
- Can do it smartly based on browser information
- Launch information gathering modules immediately
- Attacker can use the information for more targeted attacks

Putting It Together

1. Browser requests page
2. Shank downgrades encoding
3. Web server response
4. Shank Injects hook
5. Browser hooked by BeEF
6. Autorun launches modules



Initial Information Gathering

- Fingerprint plugins for targeted exploit
- Gather browser information
- Current page views
- Enable key logging
- Send to NTLM capture modules for MSF
- Run persistence modules

Demos

More Information

- Browser Exploitation Framework
  - URL: http://www.beefproject.com/
  - Rest API Information: https://github.com/beefproject/beef/wiki/BeEF-RESTful-API
- Ettercap
  - URL: http://ettercap.sourceforge.net/
- Cain & Abel
  - URL: http://www.oxid.it/cain.html

Code / Contact Info

Shank and other scripts on github.com/spiderLabs

- Available today!
- Come on up for signed CD's

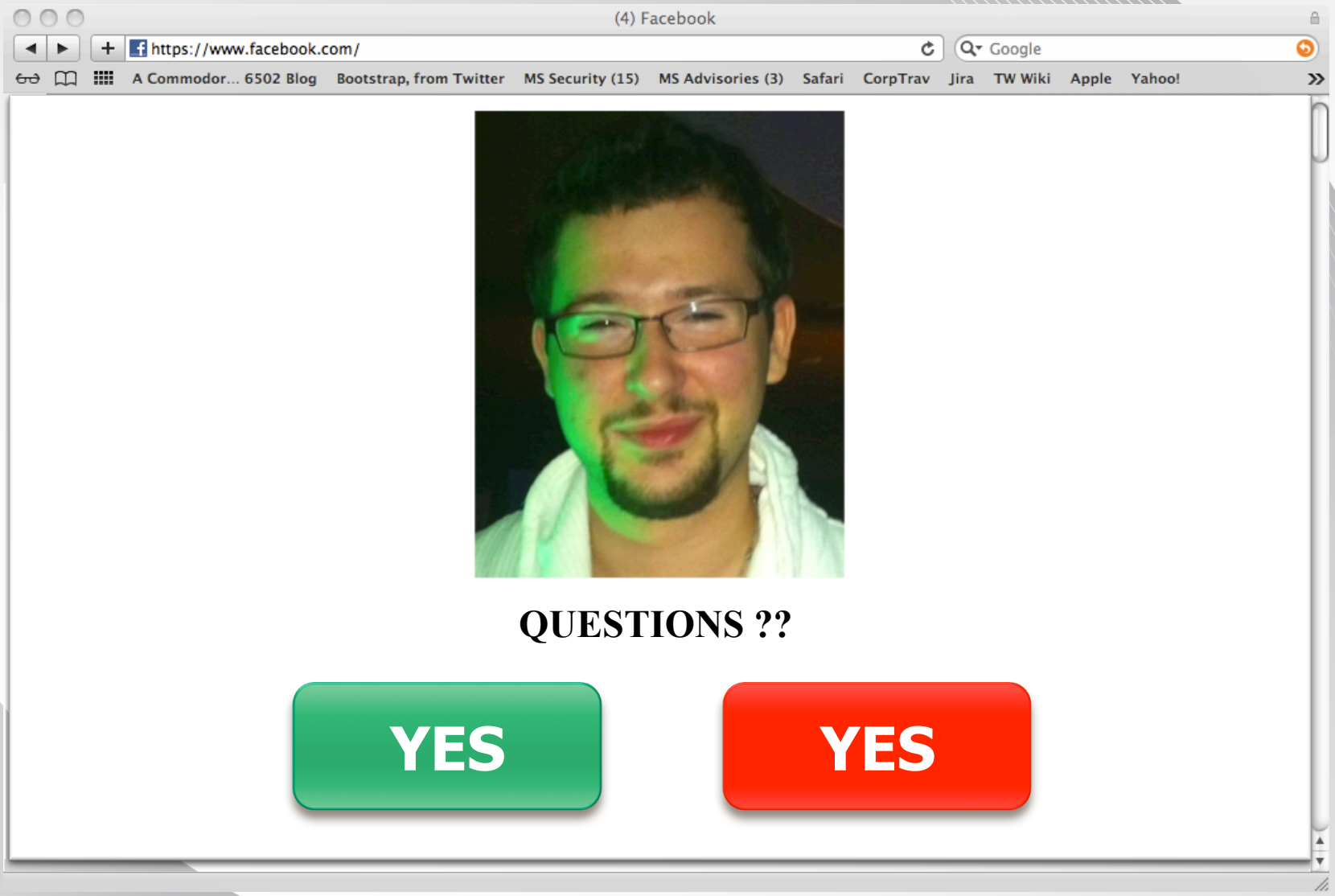
Ryan Linn
@sussurro
rlinn@trustwave.com

Steve Ocepek
@nosteve
socepek@trustwave.com

Big thanks to other SL folks

- Michele Orru, BeEF project lead, for ideas and feedback.

- Mike Ryan, for fixing and rewriting my Ruby code in like 10 minutes and making me feel, super.



Trustwave SpiderLabs

Trustwave®