# Advanced M/W-ITM Techniques for Security Testers

The Intrepidus Group

# Agenda

- Who are we?
  - Jeremy Allen
  - Rajendra Umadas
- What do we do?
  - Black box Mobile App Assessments, Thick Clients, Web apps
- What Annoys Us?
  - Proxy setups for apps
  - Throwaway tools that are all similar
- "Scratching the Itch…"
  - Introducing: Mallory
  - Architecture/Design
- Demo

# Who Are We

Other than carbon-based multi-cellular life forms

# Jeremy Allen

- Principal Consultant at Intrepidus Group
- Teaches Secure Coding Principles
- Lead Mallory Developer
- OWASP, ISACA speaker

# Rajendra Umadas

- Youngin' of the group
- Cisco CCNA Networking → CS → CompE → Computer security →Bar → Intrepidus Group
- Mobile Application Security == fun
  - Legacy Bugs
- First (of many) Black Hat Presentation

# What We Do

# Targets

- Mobile App Assessments
  - QUALCOMM/BREW
  - RIM
  - Windows Mobile
  - iPhone
  - Android
- Web Application Assessments
  - X$$ -- It pays the bills.
- Blackboxy-stuff
  - Thick client apps/plugins that talk somehow
  - Binary protocols

# Targets

- Mobile Applications
  - Often just a thick client using HTTP to transport data
- Thick Clients
  - Often just software using HTTP to transport data
- Web Applications
  - Often just browser based apps using HTTP to transport data.

# Often is not Always!

- The above targets sometimes use proprietary transport protocols.
- It may be hard to force a thick client or mobile app to tunnel data through HTTP proxy even if they use HTTP for transport
- Web Apps may use ActiveX, Flash, or various other modules that add non-standard traffic

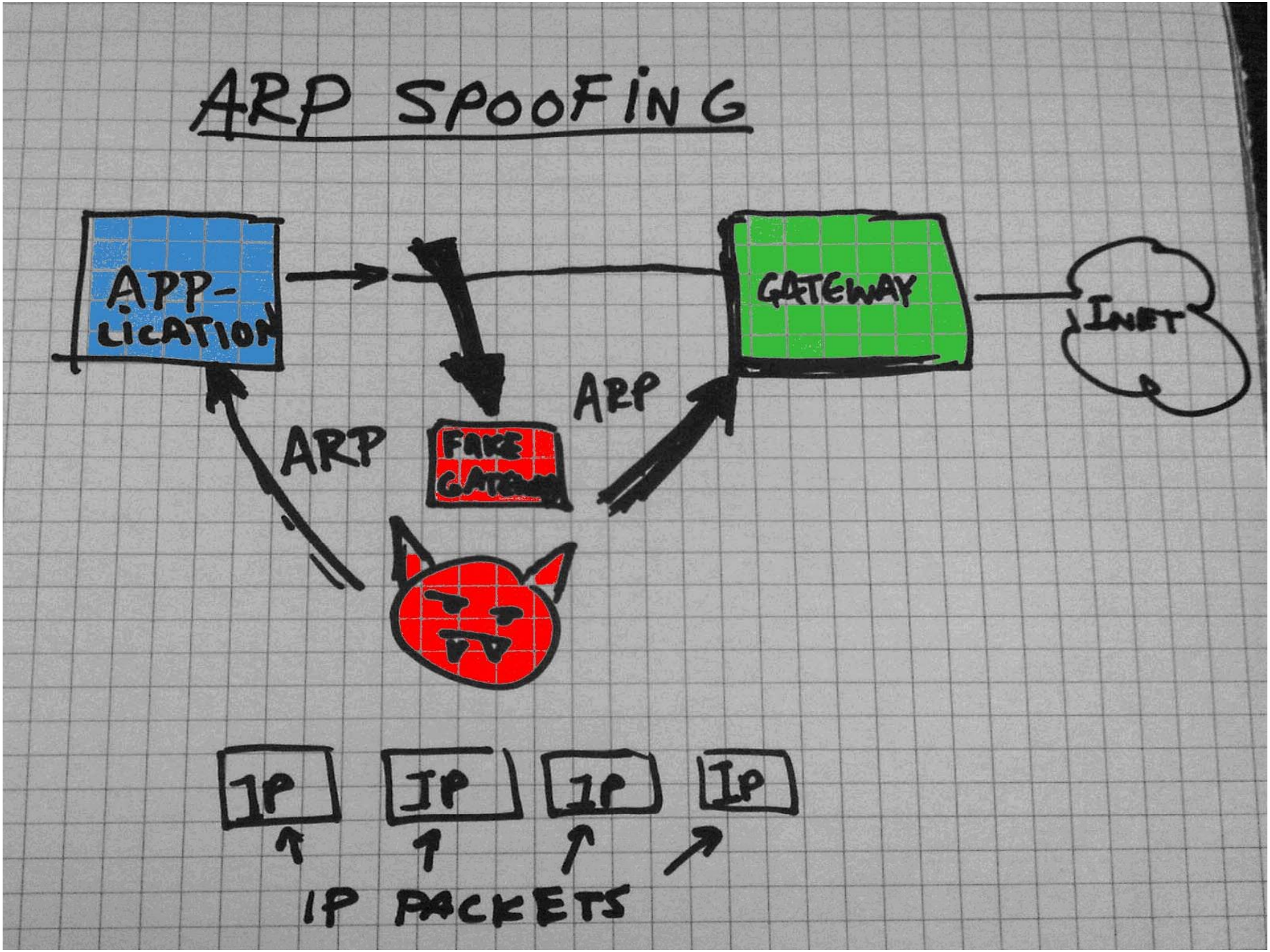
# What Annoys Us?

# Problems

- Binary protocols
- Non-proxy aware apps
- TCP reassembly from packet captures
- “Roll your own” clients
- It’s not using HTTP.. what is it ?
- Throwaway code per-engagement
- Someone else wrote the proxy and we can’t change the code easily

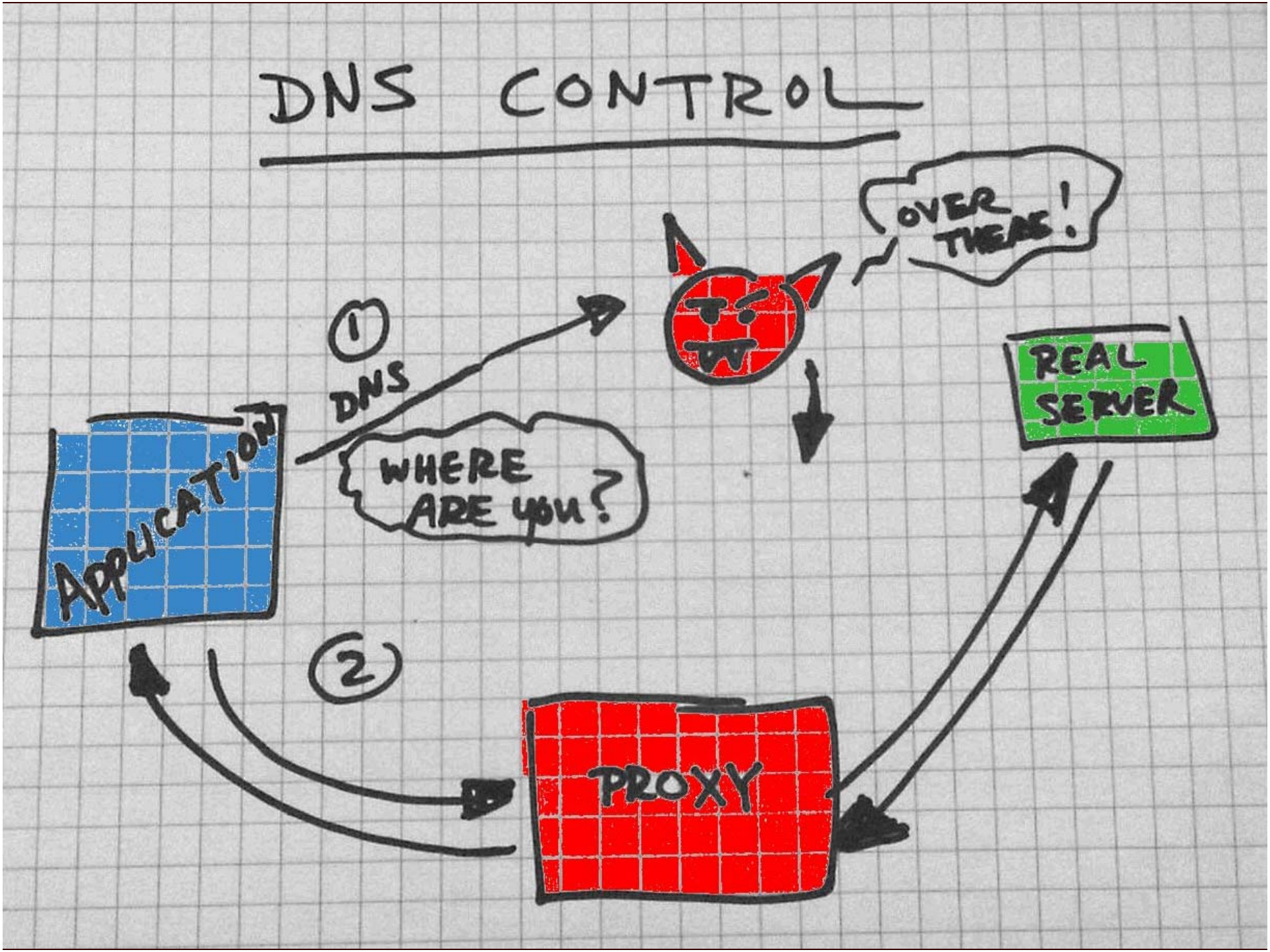
# ARP Spoofing

- Cain and Able, Ettercap…
- Limitations:
  - Dealing with IP Packets
  - Not working with TCP Streams
  - Building from Layer 3 and up???
  - Not  optimal for application assessments
  - Optimal for messing with the office

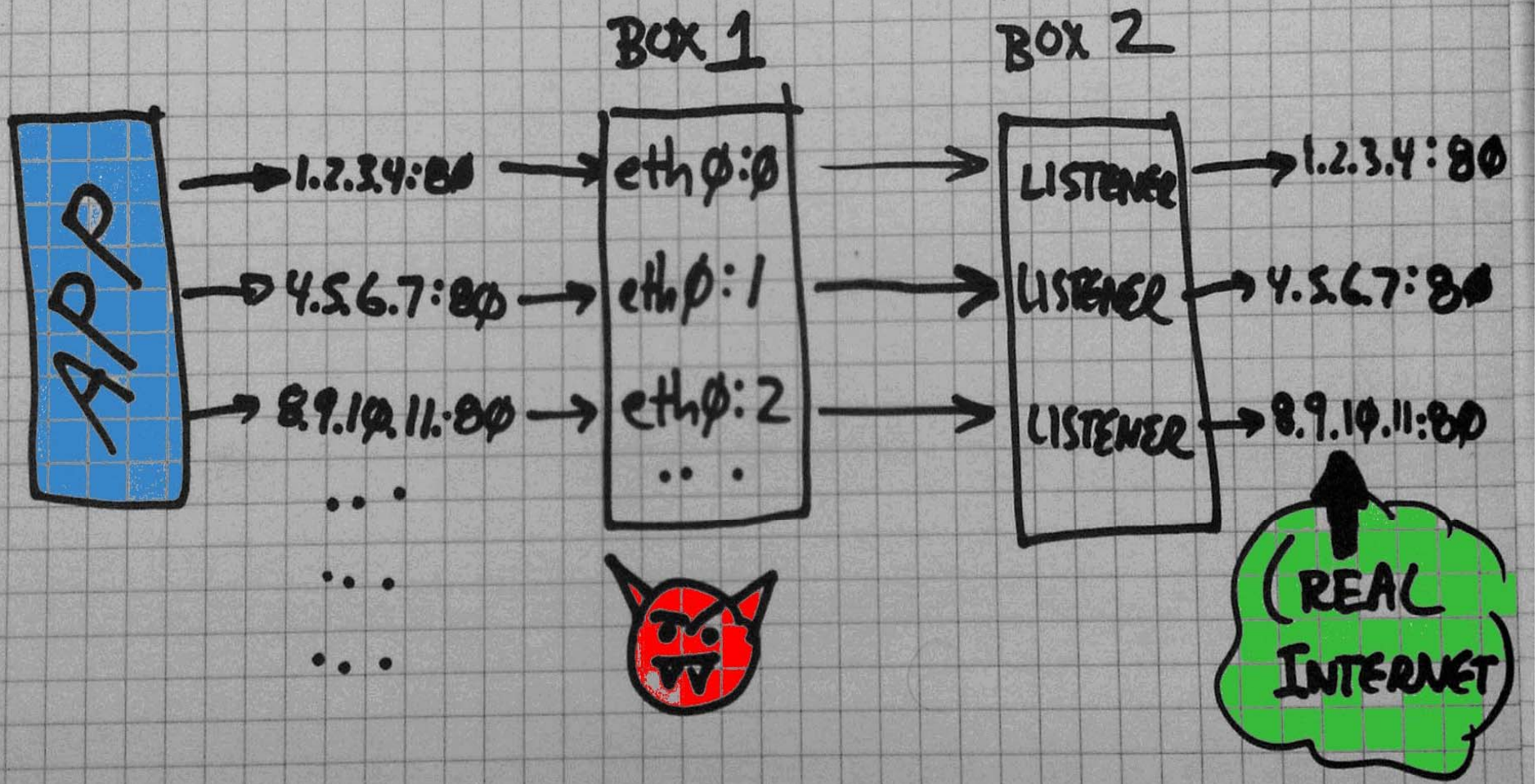

ARP SPOOFING
APP-LICATION
GATEWAY
INET
ARP
ARP
FAKE GATE
IP
IP
IP
IP
IP PACKETS

# Controllable DNS Server

- http://code.activestate.com/recipes/491264-mini-fake-dns-server/, Host File Edit
- Control Application Data Endpoints
  - You tell application who to talk to
- Application don't respect DNS/Host file
- Can't set DNS server
- Multiple Host Same Port?!
  - Ifconfig aliased?!


DNS CONTROL
OVER THERE!
1
DNS
WHERE ARE YOU?
REAL SERVER
APPLICATION
2
PROXY

# IMPERSONATING REAL SERVER WITH ALIAS ETHERNET DEVICES



# Application Proxy

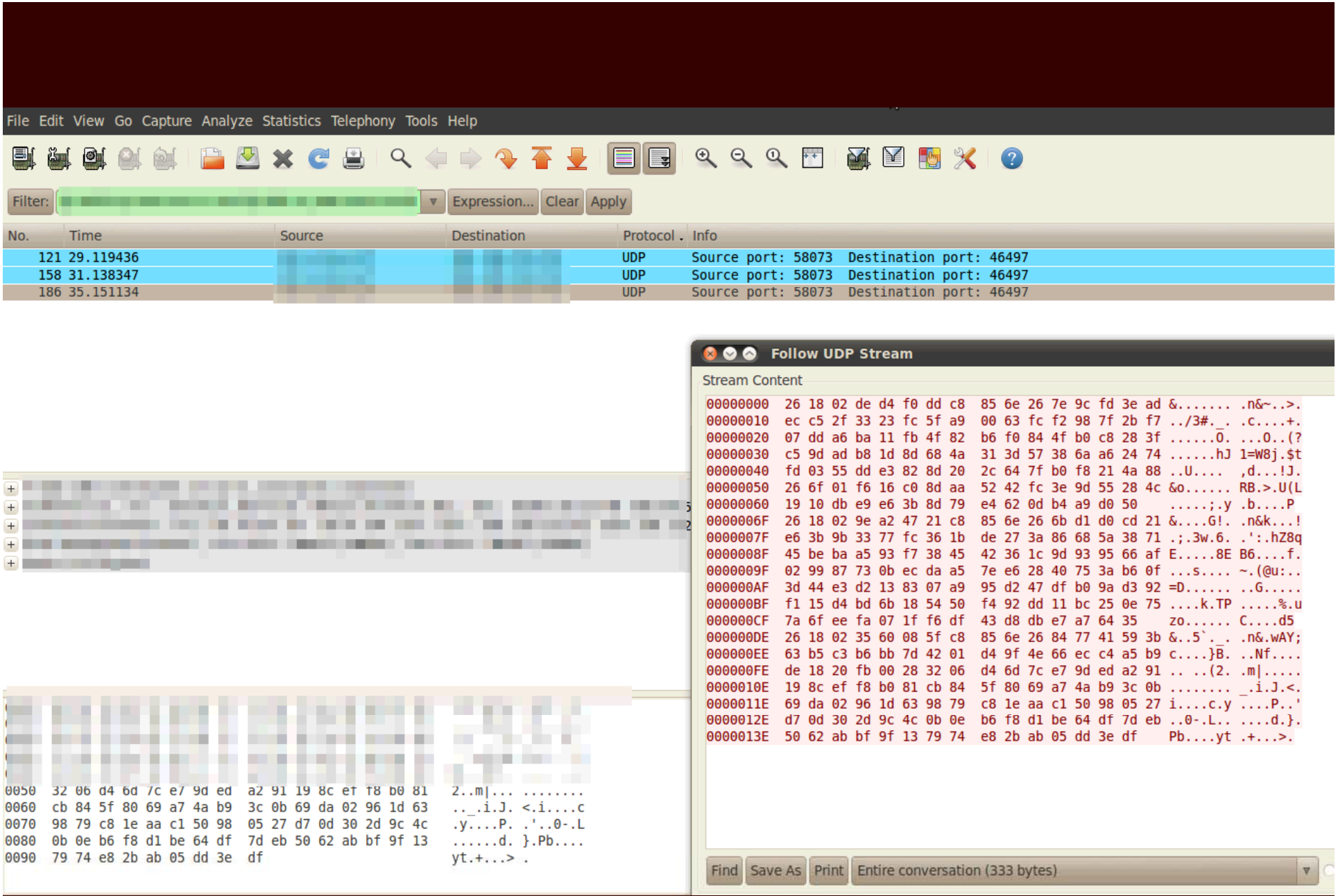
- Web Scarab, Burp, Paros…
- PFI
- Need to configure application and the proxy for full duplex communication
- Use OS TCP/IP Stack
  - No reassembly
- Limitations:
  - Application might not support proxy setting
  - Manual configuration of proxy endpoint

# Big F****** Kludge!

- When they just want to make our lives difficult
- No one tool can work with us to get the data proxied
  - Tools work well, just don't meet some specific specifications (Buffer size, platform intricacies,…)
- Time to hack it together

# Sniffers

- Gather Data
- Figure out what we need
  - Server IP(s)
  - Server Port(s)
  - TCP or UDP?
  - Buffer sizes???
  - Any other anomalous characteristics


File Edit View Go Capture Analyze Statistics Telephony Tools Help
Filter:
Expression... Clear Apply
No. Time Source Destination Protocol Info
121 29.119436 UDP Source port: 58073 Destination port: 46497
158 31.138347 UDP Source port: 58073 Destination port: 46497
186 35.151134 UDP Source port: 58073 Destination port: 46497
Follow UDP Stream
Stream Content
00000000 26 18 02 de d4 f0 dd c8 85 6e 26 7e 9c fd 3e ad &..... .n&~..>.
00000010 ec c5 2f 33 23 fc 5f a9 00 63 fc f2 98 7f 2b f7 ../3#._. .c....+.
00000020 07 dd a6 ba 11 fb 4f 82 b6 f0 84 4f b0 c8 28 3f ......O. ...O..(?
00000030 c5 9d ad b8 1d 8d 68 4a 31 3d 57 38 6a a6 24 74 ......hJ 1=W8j.$t
00000040 fd 03 55 dd e3 82 8d 20 2c 64 7f b0 f8 21 4a 88 ..U.... ,d...!J.
00000050 26 6f 01 f6 16 c0 8d aa 52 42 fc 3e 9d 55 28 4c &o..... RB.>.U(L
00000060 19 10 db e9 e6 3b 8d 79 e4 62 0d b4 a9 d0 50 .....;.y .b....P
0000006F 26 18 02 9e a2 47 21 c8 85 6e 26 6b d1 d0 cd 21 &....G!. .n&k...!
0000007F e6 3b 9b 33 77 fc 36 1b de 27 3a 86 68 5a 38 71 .;.3w.6. .':.hZ8q
0000008F 45 be ba a5 93 f7 38 45 42 36 1c 9d 93 95 66 af E.....8E B6....f.
0000009F 02 99 87 73 0b ec da a5 7e e6 28 40 75 3a b6 0f ...s.... ~.(@u:..
000000AF 3d 44 e3 d2 13 83 07 a9 95 d2 47 df b0 9a d3 92 =D..... ..G.....
000000BF f1 15 d4 bd 6b 18 54 50 f4 92 dd 11 bc 25 0e 75 ....k.TP .....%.u
000000CF 7a 6f ee fa 07 1f f6 df 43 d8 db e7 a7 64 35 zo..... C....d5
000000DE 26 18 02 35 60 08 5f c8 85 6e 26 84 77 41 59 3b &..5`._. .n&.wAY;
000000EE 63 b5 c3 b6 bb 7d 42 01 d4 9f 4e 66 ec c4 a5 b9 c....}B. ..Nf....
000000FE de 18 20 fb 00 28 32 06 d4 6d 7c e7 9d ed a2 91 .. ..(2. .m|.....
0000010E 19 8c ef f8 b0 81 cb 84 5f 80 69 a7 4a b9 3c 0b ........ _.i.J.<.
0000011E 69 da 02 96 1d 63 98 79 c8 1e aa c1 50 98 05 27 i....c.y ....P..'
0000012E d7 0d 30 2d 9c 4c 0b 0e b6 f8 d1 be 64 df 7d eb ..0-.L.. ....d.}.
0000013E 50 62 ab bf 9f 13 79 74 e8 2b ab 05 dd 3e df Pb....yt .+...>.
Find Save As Print Entire conversation (333 bytes)
0050 32 06 d4 6d 7c e7 9d ed a2 91 19 8c ef f8 b0 81 2..m|... ........
0060 cb 84 5f 80 69 a7 4a b9 3c 0b 69 da 02 96 1d 63 .._.i.J. <.i....c
0070 98 79 c8 1e aa c1 50 98 05 27 d7 0d 30 2d 9c 4c .y....P. .'..0-.L
0080 0b 0e b6 f8 d1 be 64 df 7d eb 50 62 ab bf 9f 13 ......d. }.Pb....
0090 79 74 e8 2b ab 05 dd 3e df yt.+...> .

# Routing/NATting/Mangling

- -A PREROUTING -i eth1 -p tcp -m tcp --dport 80 -j REDIRECT --to-ports 1231
- -A PREROUTING -i eth1 -p tcp -m tcp --dport 443 -j REDIRECT --to-ports 1232
- -A PREROUTING -i eth1 -p tcp -m tcp --dport 4356-j REDIRECT --to-ports 1233

# *cat

- netcat
  - nc -l -p 1231 | nc host 80
  - nc -l -p 1232 | nc host 443
  - nc -l -p 1233 | nc host 4356
- socat
  - socat -v -v -x tcp-listen:80  host:80

# Scripting Languages

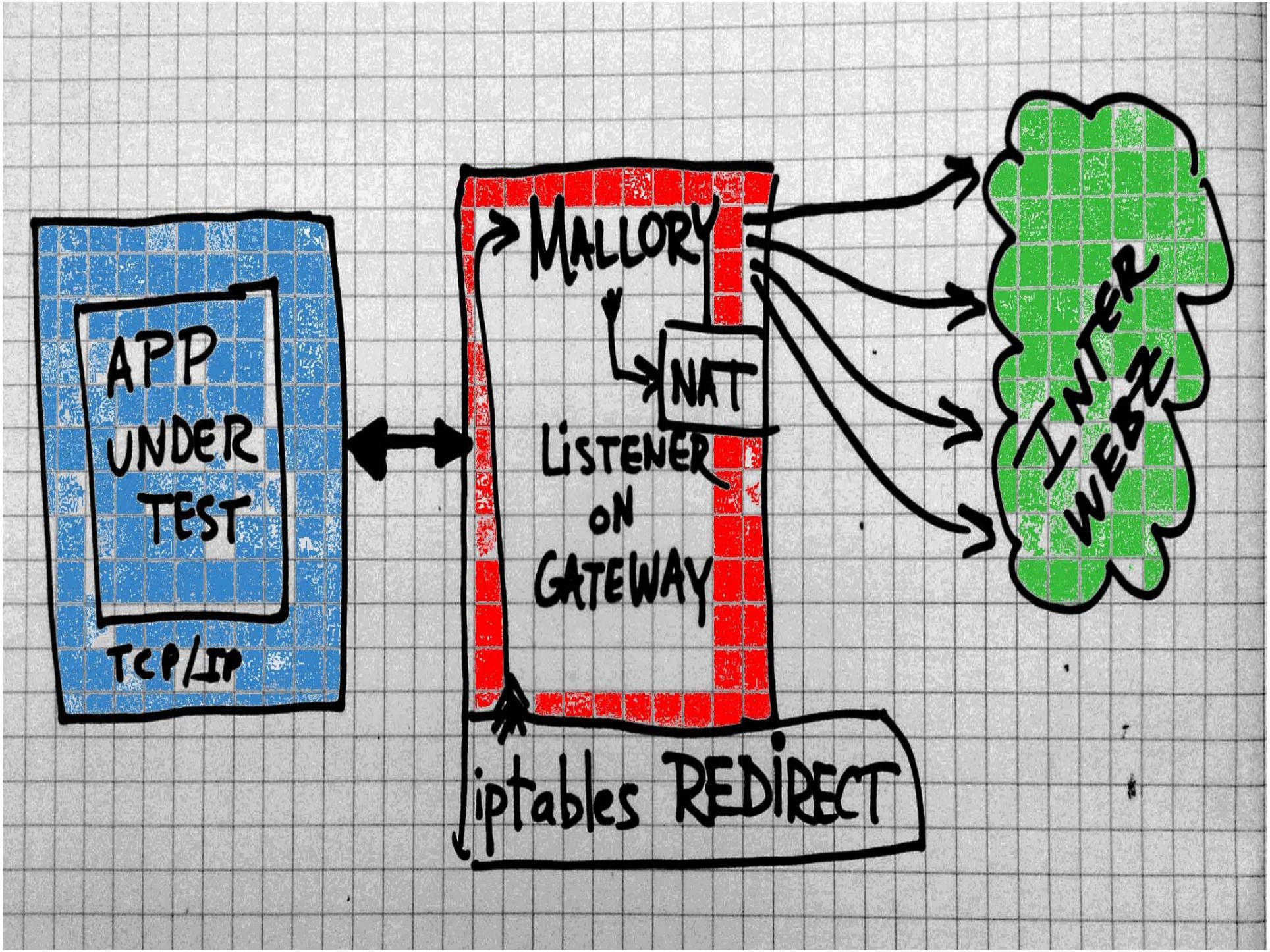
- Python to recreate client from recon above
- Python:
  - Open local listening socket
  - Read data forward to socket
  - Manipulate data read from local socket
  - Open remote socket to original server (from recon)
  - Forward mangled data
  - And Reverse

# Quick Comparison

| Application | Transparent Aware | Extensible | Programmatic Instream Modification | Manual Instream Modification | Non-HTTP |
|---|---|---|---|---|---|
| Mallory | X | X | X | X | X |
| Burp (HTTP Proxies) | | | | X | |
| PFI | | X | | X | X |
| Cain and Abel | | | | | X |
| Middler | | X | X | | |
| Ethercap | | X | X | | X |
| Netsed | | | X | | X |
| Squid | X | * | X | | |

# “Scratching the Itch…”

<redacted>


APP UNDER TEST
TCP/IP
MALLORY
NAT
LISTENER ON GATEWAY
iptables REDIRECT
INTER WEBZ

# Mallory: Guts / Dependencies

- Python2.6, python2.6-dev
- python-setuptools
- python-pyasn1
- python-netfilter
- libnetfilter-conntrack-dev
- pynetfilter_conntrack
- netfilter-extensions-source
- libnetfilter-conntrack3-dbg
- libnetfilter-conntrack1_0.0.99-1_i386.deb
- python-paramiko
- (http://software.inl.fr/trac/wiki/pynetfilter_conntrack)

# Mallory: Proxy Server Server

- mallory.py starts a listener (SERVER)
- This server accepts() connections
- Each accept determines endpoint source (IP, Port)
- Integrates with netfilter to determine the "Original destination" (pre-REDIRECT)
- Creates 2 proxy threads (forward/reverse) to shovel traffic back and forth
- Plugins are called in the middle of the shovel

# Mallory: Directory Structure

We're using "hg" (mercurial) on bitbucket to store code/track changes.

After hg checkout,

- ca/ : Where dynamic certs are stored
- db/ : Where traffic database files live
- src/ : Executable python code/plugins

# Mallory: Command Line Opts

## *-n, --no-transparent*

Turn off the transparent proxy. This puts the proxy into a static only mode. You must supply the destination IP:Port you want traffic to go to

**Example: -n ip:port, -n a.b.c.d:443**

## *-l, --listen*

Specify the port to listen on.
Only useful with --no-transparent. Format is:
**example: -l 443**

Default is 20755

# Mallory: Command Line Opts(2)

## *-d / –trafficdb*

Specify the traffic database name, default is "trafficdb" SQLITE databases are used.

## *-p / --proto*

Specify the protocol to use. This is only useful with the no-transparent option. You must specify the module inside of the protocol package as well as the exact protocol class to instantiate.

**Example:** -p sslproto.SSLProtocol or -p ssh.SSHProtocol
or -p http.HTTP

# Mallory: Database Schema

## *TABLE: connections*

connCount INTEGER – Uniquely identifies a TCP Stream in this database.
Usedto JOIN on from flows table
serverIp TEXT,serverPort INTEGER : Original destination where packet is going (pre-REDIRECT)
clientIp TEXT, clientPort INTEGER (Victim client)

## *TABLE: flows*

Actual data in connections
Buffer sizes of real connection are preserved.

conncount INTEGER: Unique ID for this flow maps to connections
direction TEXT,: "c2s" / "s2c"
buffindex INTEGER: Increasing number for this stream
timestamp FLOAT,: Seconds since epoch.
buffer BLOB: The actual data received.

# Mallory: Database Schema (2)

*TABLE: dgram* : for UDP "connectionless" data

saddr : IP Address in normal dotted quad format
sport: Integer source port

daddr : IP Addres in dotted quad

dport INTEGER:destination port

direction TEXT : "c2s" or "s2c"

body BLOB: The actual data contained in the packet
timestamp: seconds since the epoch

# Mallory: User Interfaces (CLI)

- CLI -> Command Line Interface
- Connects to Mallory server
- Some commands (WIP):

  *[a] auto send mode*

  *[m] manual mode*

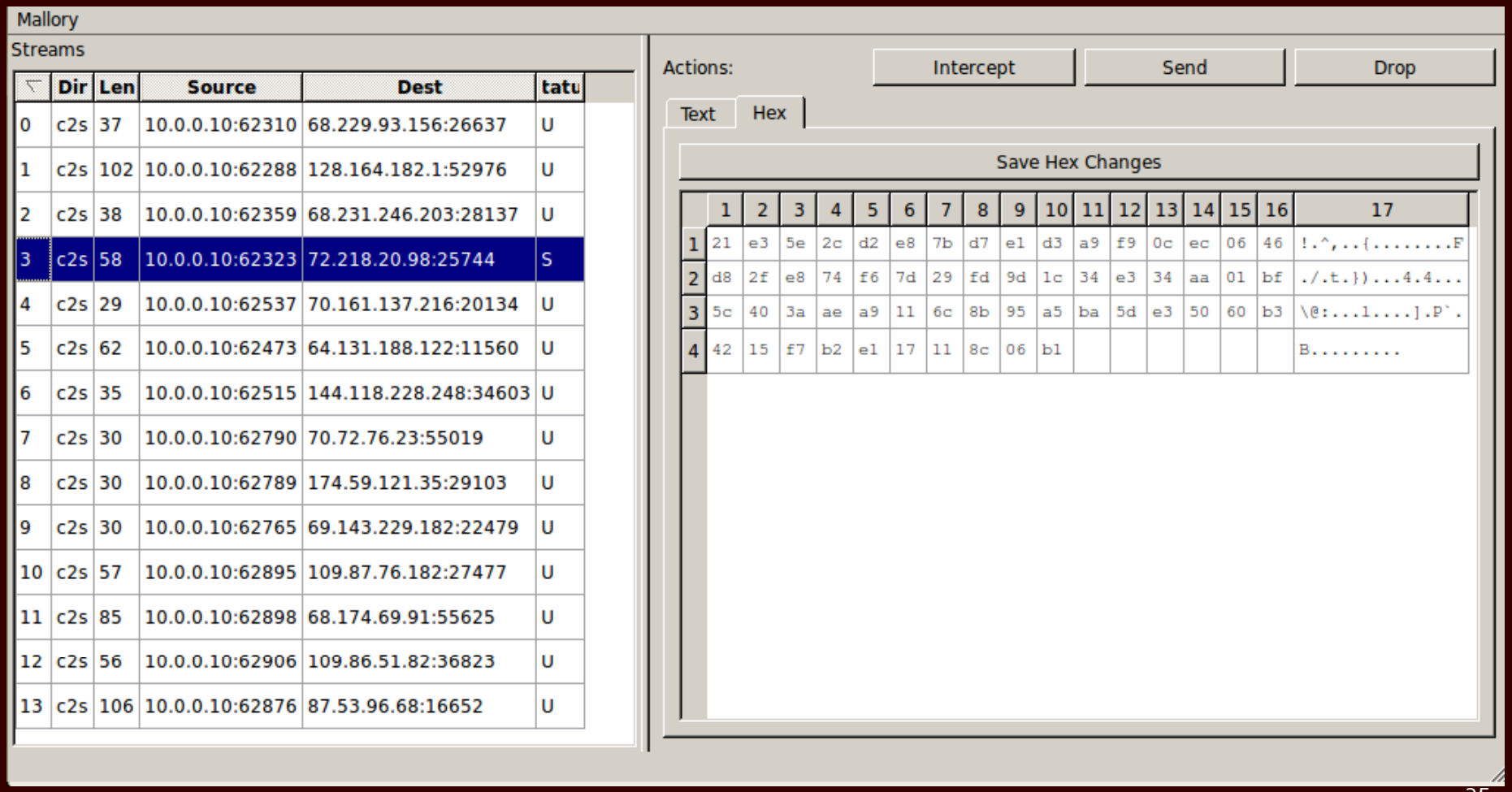
  *[o] debugger off at server*

  *[n] debugger on at server*

  *[q] quit*

# Mallory: Graphical Interface

- Uses same XMLRPC mechanism to talk to server




| | Dir | Len | Source | Dest | tatu |
|---|---|---|---|---|---|
| 0 | c2s | 37 | 10.0.0.10:62310 | 68.229.93.156:26637 | U |
| 1 | c2s | 102 | 10.0.0.10:62288 | 128.164.182.1:52976 | U |
| 2 | c2s | 38 | 10.0.0.10:62359 | 68.231.246.203:28137 | U |
| 3 | c2s | 58 | 10.0.0.10:62323 | 72.218.20.98:25744 | S |
| 4 | c2s | 29 | 10.0.0.10:62537 | 70.161.137.216:20134 | U |
| 5 | c2s | 62 | 10.0.0.10:62473 | 64.131.188.122:11560 | U |
| 6 | c2s | 35 | 10.0.0.10:62515 | 144.118.228.248:34603 | U |
| 7 | c2s | 30 | 10.0.0.10:62790 | 70.72.76.23:55019 | U |
| 8 | c2s | 30 | 10.0.0.10:62789 | 174.59.121.35:29103 | U |
| 9 | c2s | 30 | 10.0.0.10:62765 | 69.143.229.182:22479 | U |
| 10 | c2s | 57 | 10.0.0.10:62895 | 109.87.76.182:27477 | U |
| 11 | c2s | 85 | 10.0.0.10:62898 | 68.174.69.91:55625 | U |
| 12 | c2s | 56 | 10.0.0.10:62906 | 109.86.51.82:36823 | U |
| 13 | c2s | 106 | 10.0.0.10:62876 | 87.53.96.68:16652 | U |

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 21 | e3 | 5e | 2c | d2 | e8 | 7b | d7 | e1 | d3 | a9 | f9 | 0c | ec | 06 | 46 | !.^,..{.....F |
| 2 | d8 | 2f | e8 | 74 | f6 | 7d | 29 | fd | 9d | 1c | 34 | e3 | 34 | aa | 01 | bf | ./.t.})...4.4... |
| 3 | 5c | 40 | 3a | ae | a9 | 11 | 6c | 8b | 95 | a5 | ba | 5d | e3 | 50 | 60 | b3 | \@:...l....].P`. |
| 4 | 42 | 15 | f7 | b2 | e1 | 17 | 11 | 8c | 06 | b1 | | | | | | | B..... |

# Mallory: Plug-in Code

- An API !
- Base classes to inherit from and build custom protocols with:
  TcpProtocol, UdpProtocol
- HttpProtocol Implementation -- fairly complete – it's a mod of the Python one☺
- A simple "event" protocol for plugins to register interest and call handlers on:

CREATE, ACCEPT, and FORWARDING EVENTS

# Demo

Get up and stretch!!!!