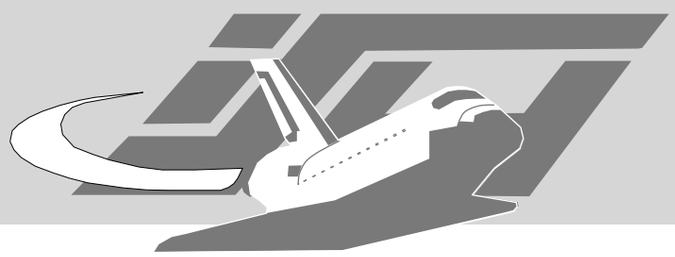
BLACK HAT USA 2005

Hands on Honeypot Technology

honeyd

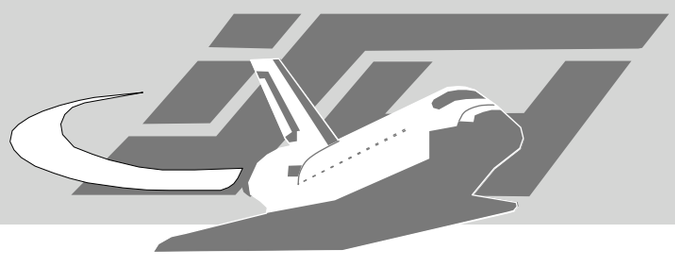
Maximillian Dornseif
Thorsten Holz

RHEINISCH-
WESTFÄLISCHE
TECHNISCHE
HOCHSCHULE
AACHEN
RWTH
UNIVERSITY
OF TECHNOLOGY
AACHEN



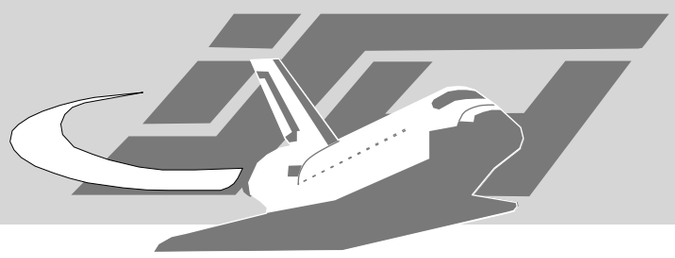
Schedule

	Monday	**Tuesday**
9:00 10:30	Intro	forensics
10:45 12:30	Gen II/III Honeynets	botnets
14:00 16:00	honeyd distributed honeynets	attacks
16:15 18:00	nwcollect nephentis	leurre honeyfarms outlook

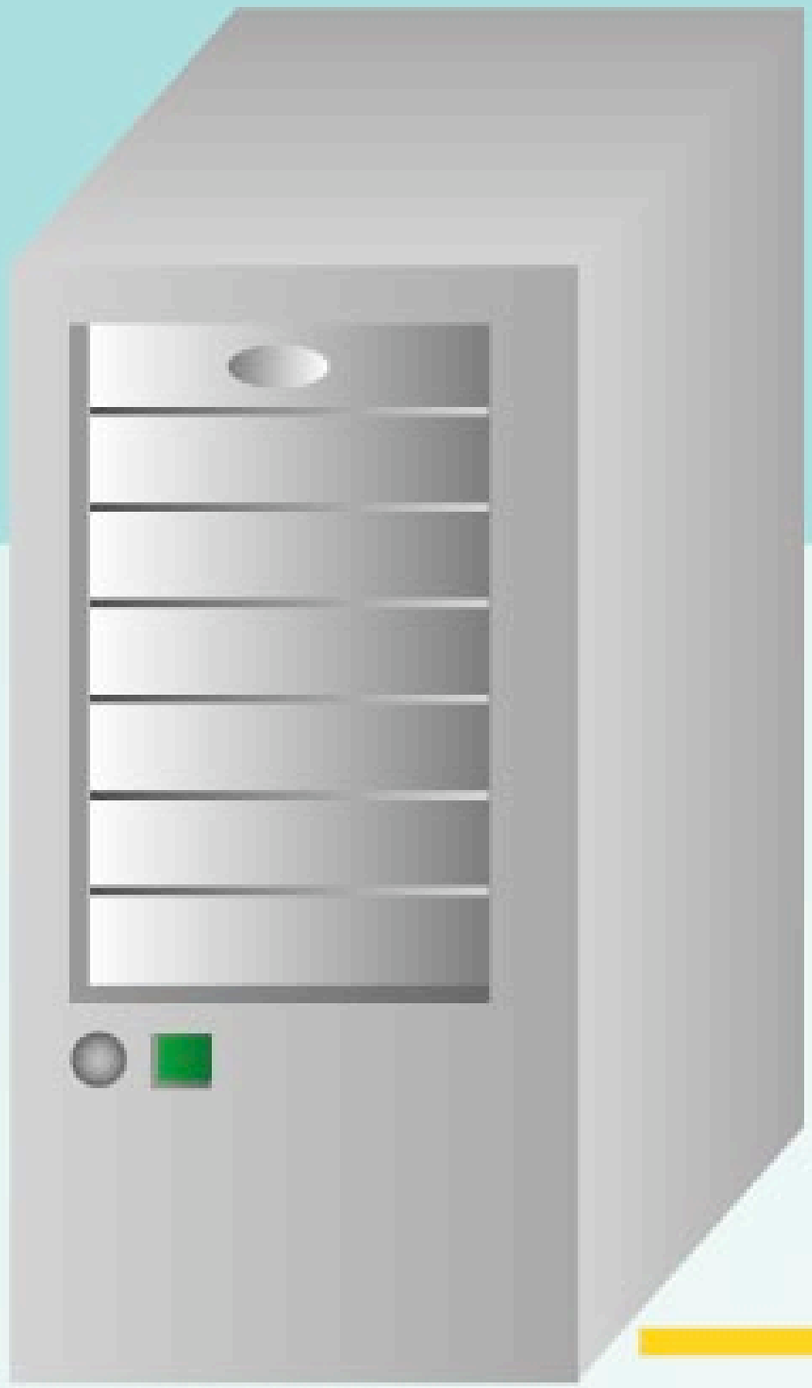


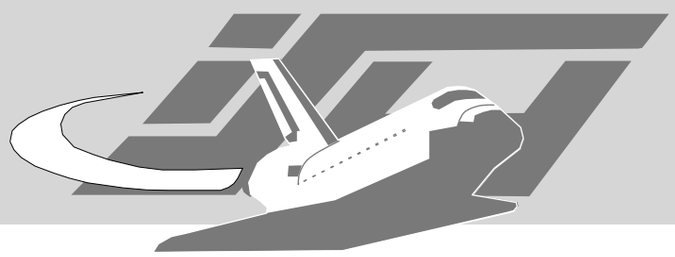
Outline

- Introduction into honeyd
- Getting packets to honeyd
- honeyd configuration
- HOACD
- gathering results

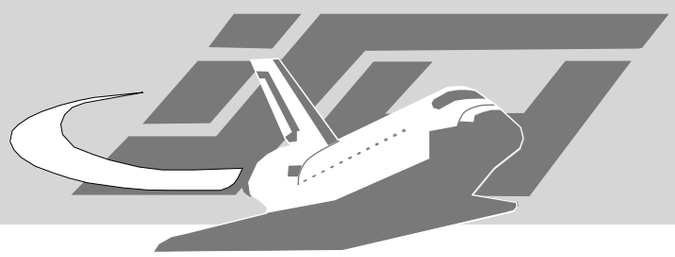


honeyd

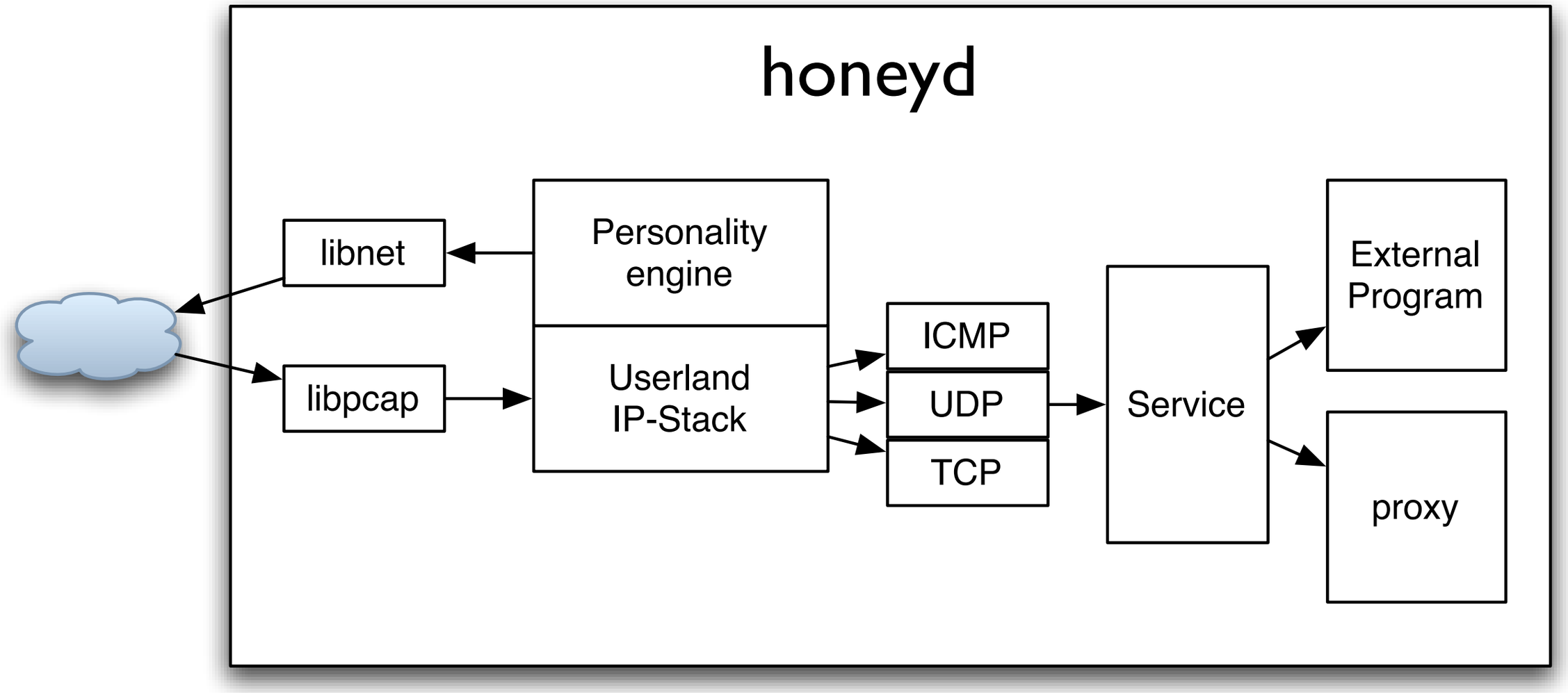




- honeyd is a GPLed pice of software by Nils Provos
  - http://www.honeyd.org/
- An engine for running many virtual IP-stacks in parallel
- http://niels.xtdnet.nl/papers/honeyd.pdf
- BQS?

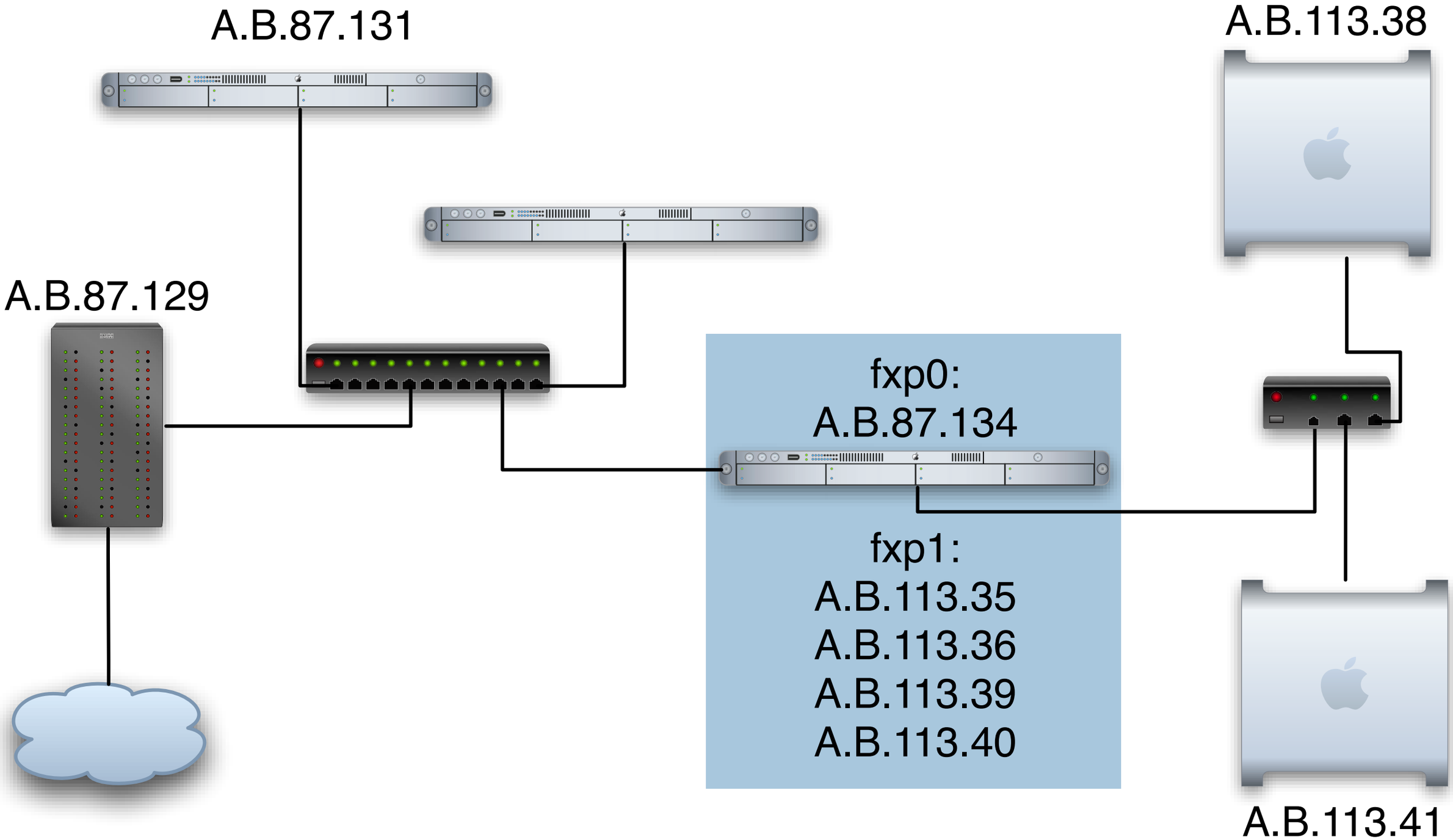


- An engine for running many virtual IP-stacks in parallel
  - includes routing
  - can emulate the pekularities from many IP-stacks
    - super-smart: uses the os fingerprint databases of nmap, Xprobe and p0f as the basis for emulation
- Version 1.0 includes graphics, a webserver, central data collection ...
- Version 2.0 will probably be able to read mail


honeyd
libnet
libpcap
Personality engine
Userland IP-Stack
ICMP
UDP
TCP
Service
External Program
proxy

Getting Data to honeyd

Our example Net

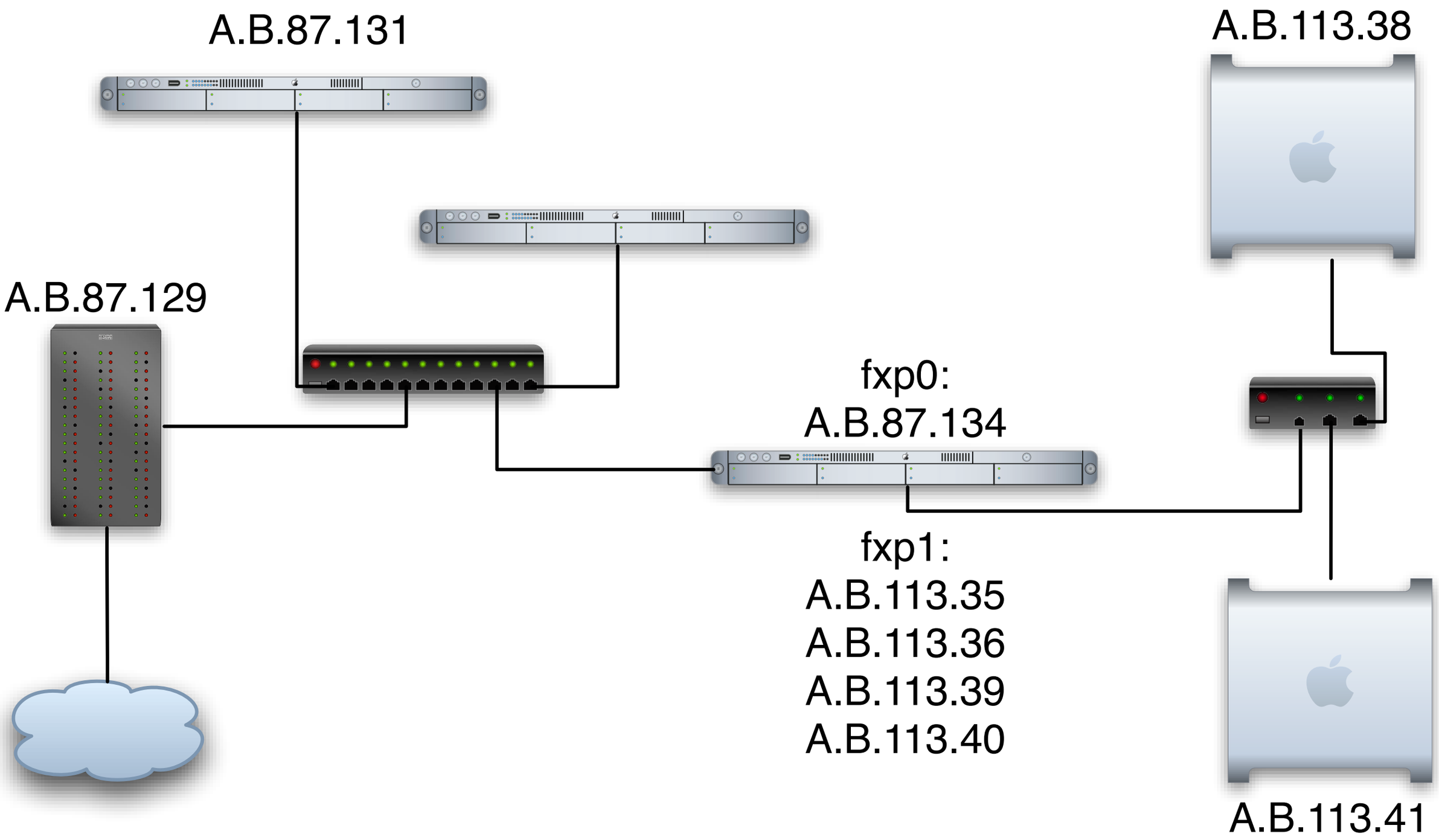


Part 1:
using OS means/arp

Terminal – RedTeam@RWTH

```
md@beebop:~ $ sudo honeyd -d -P A.B.87.133
Honeyd V1.0 Copyright (c) 2002-2004 Niels Provos
honeyd[90527]: started with -d -P A.B.87.133
Warning: Impossible SI range in Class fingerprint "IBM OS/400 V4R2M0"
Warning: Impossible SI range in Class fingerprint "Microsoft Windows NT
4.0 SP3"
honeyd[90527]: listening promiscuously on fxp0: (arp or ip proto 47 or
(udp and src port 67 and dst port 68) or (ip and (host A.B.87.133))) and
not ether src 00:d0:b7:c6:0f:a7
honeyd[90527]: switching to polling mode
honeyd[90527]: Demoting process privileges to uid 32767, gid 32767
```

Our example Net



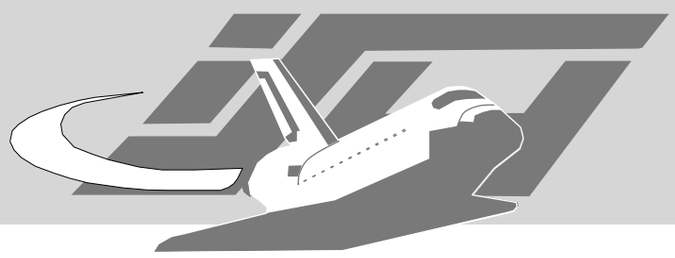
Terminal – RedTeam@RWTH

```
md@beebop:~ $ sudo tcpdump -n arp
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on fxp0, link-type EN10MB (Ethernet), capture size 96 bytes
15:41:02.791113 arp who-has A.B.87.133 tell A.B.87.129
15:41:09.016834 arp who-has A.B.87.133 tell A.B.87.129
15:41:10.018347 arp who-has A.B.87.133 tell A.B.87.129

$ sudo arp -s A.B.87.133 auto pub
using interface fxp0 for proxy with address 00:d0:b7:c6:0f:a7
```

Terminal – RedTeam@RWTH

```
15:48:37.269737 arp who-has A.B.87.133 tell A.B.87.129
15:48:37.269786 arp reply A.B.87.133 is-at 00:d0:b7:c6:0f:a7

honeyd[90527]: Connection to closed port: udp (216.148.212.182:19870 - A.B.87.133:53)
honeyd[90527]: Connection to closed port: udp (216.148.212.182:17078 - A.B.87.133:53)
honeyd[90527]: Connection to closed port: udp (216.148.212.182:38895 - A.B.87.133:53)
honeyd[90527]: Connection to closed port: udp (216.148.212.182:57440 - A.B.87.133:53)
honeyd[90527]: Connection to closed port: udp (217.19.192.139:33038 - A.B.87.133:53)
honeyd[90527]: Connection to closed port: udp (216.148.212.182:39476 - A.B.87.133:53)
honeyd[90527]: Connection to closed port: udp (216.148.212.182:40954 - A.B.87.133:53)
honeyd[90527]: Connection to closed port: udp (216.148.212.182:43450 - A.B.87.133:53)
honeyd[90527]: Connection to closed port: udp (216.148.212.182:16194 - A.B.87.133:53)
honeyd[90527]: Connection to closed port: udp (216.148.212.182:63706 - A.B.87.133:53)
honeyd[90527]: Connection to closed port: udp (216.148.212.182:11662 - A.B.87.133:53)
honeyd[90527]: Connection to closed port: udp (216.148.212.182:10994 - A.B.87.133:53)
```

Part 2:
using brute force/arpd



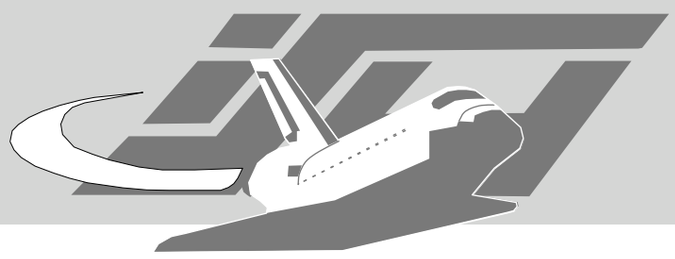
Several arpd's

- Nils Provos' arpd
  - http://www.citi.umich.edu/u/provos/honeyd/arpd-0.2.tar.gz
- Alexey Kuznetsov's arpd from iproute
  - you don't want this
- farpd
  - that's Nils Provos' arpd repackaged by Debian.

Ein offenes Betriebssystem kann schon mal mutieren. Bei Windows 2000 hingegen gibt es alle Services und Dienste aus einer Hand. Das spart Zeit und somit wirklich Geld. Mehr Infos unter www.microsoft.com/germany/windows2000

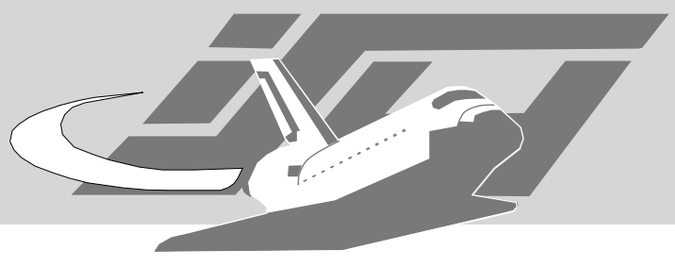
Microsoft

ein offenes betriebssystem hat nicht nur vorteile



- arpd replies to any ARP request for an IP address matching the specified destination net with the hardware MAC address of the specified interface, but only after determining if another host already claims it.
- Any IP address claimed by arpd is eventually forgotten after a period of inactivity or after a hard timeout, and is relinquished if the real owner shows up.
- This enables a single host to claim all unassigned addresses on a LAN for network monitoring or simulation.

arpd-0.2 manpage



- arpd [-d] [-i interface] [net ...]
- net

  The IP address or network (specified in CIDR notation) or IP address ranges to claim (e.g. ``10.0.0.3'', ``10.0.0.0/16'' or ``10.0.0.5-10.0.0.15''). If unspecified, farpd will attempt to claim any IP address it sees an ARP request for. Mutiple addresses may be specified.

arpd-0.2 manpage

Terminal – RedTeam@RWTH

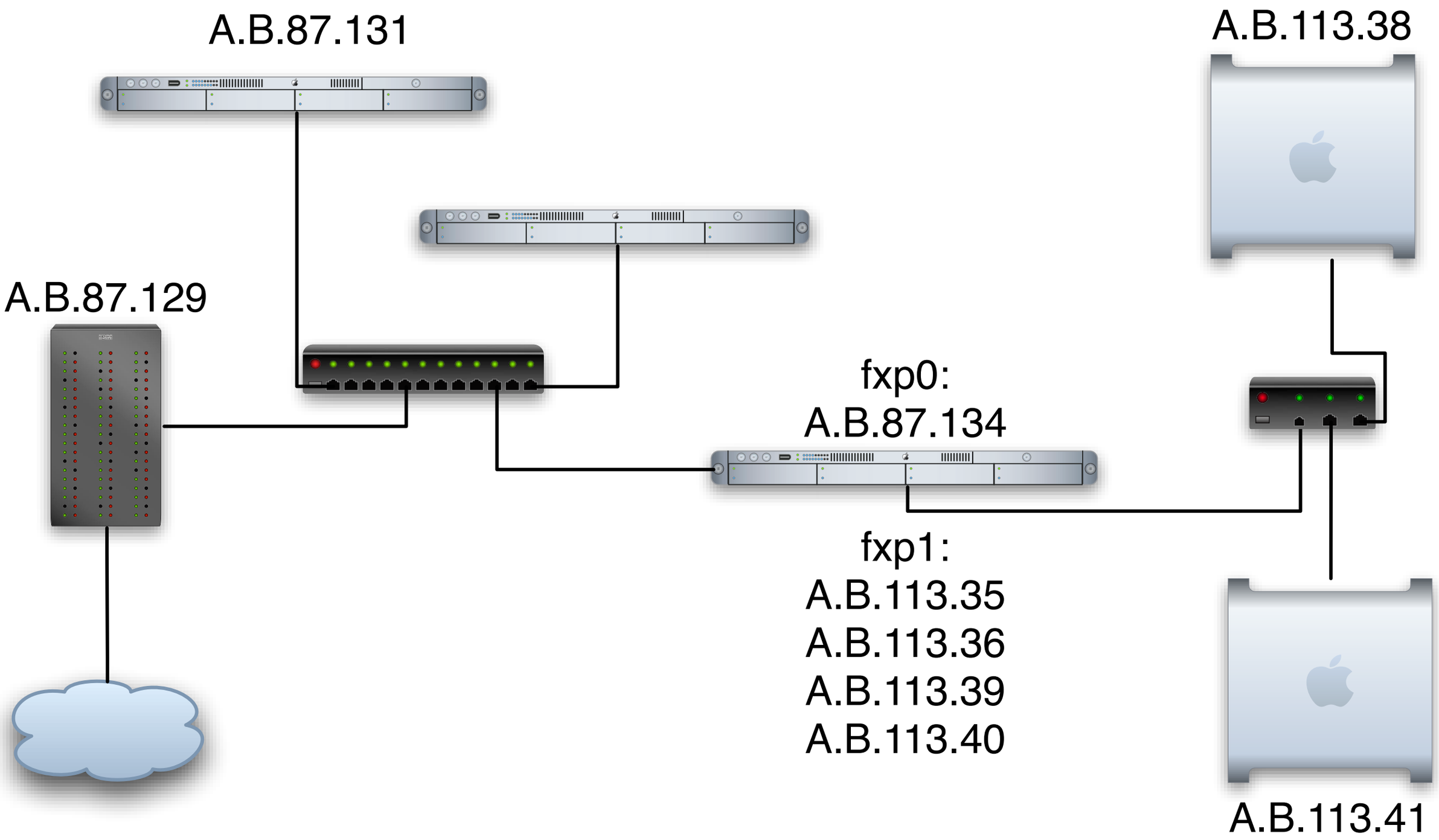
```
md@beebop:~ $ sudo arpd -d A.B.87.133 A.B.87.140 A.B.87.141 A.B.
87.142 A.B.87.143 A.B.87.144 A.B.87.145
arpd[97330]: listening on fxp0: arp and (dst A.B.87.133 or dst
A.B.87.140 or dst A.B.87.141 or dst A.B.87.142 or dst A.B.87.143
or dst A.B.87.144 or dst A.B.87.145) and not ether src
00:d0:b7:c6:0f:a7
arpd[97330]: arpd_lookup: no entry for A.B.87.142
arpd[97330]: arpd_send: who-has A.B.87.142 tell A.B.87.134
arpd[97330]: arpd_lookup: no entry for A.B.87.133
arpd[97330]: arpd_send: who-has A.B.87.133 tell A.B.87.134
arpd[97330]: arpd_send: who-has A.B.87.142 tell A.B.87.134
arpd[97330]: arpd_send: who-has A.B.87.133 tell A.B.87.134
arpd[97330]: arp reply A.B.87.133 is-at 00:d0:b7:c6:0f:a7
arpd[97330]: arp reply A.B.87.133 is-at 00:d0:b7:c6:0f:a7

16:33:11.122571 arp who-has A.B.87.133 tell A.B.87.134
16:33:12.122840 arp who-has A.B.87.133 tell A.B.87.134
16:33:13.123089 arp who-has A.B.87.133 tell A.B.87.134
16:33:14.122602 arp who-has A.B.87.133 tell A.B.87.134
16:33:15.122660 arp who-has A.B.87.133 tell A.B.87.134
```

Trouble with arpd

- Might break DHCP
- Confuses hosts in various interesting ways
- **don’t** use it in product
- to slow
- can’t handle a /16 (we have tried)

Part 3:
using routing

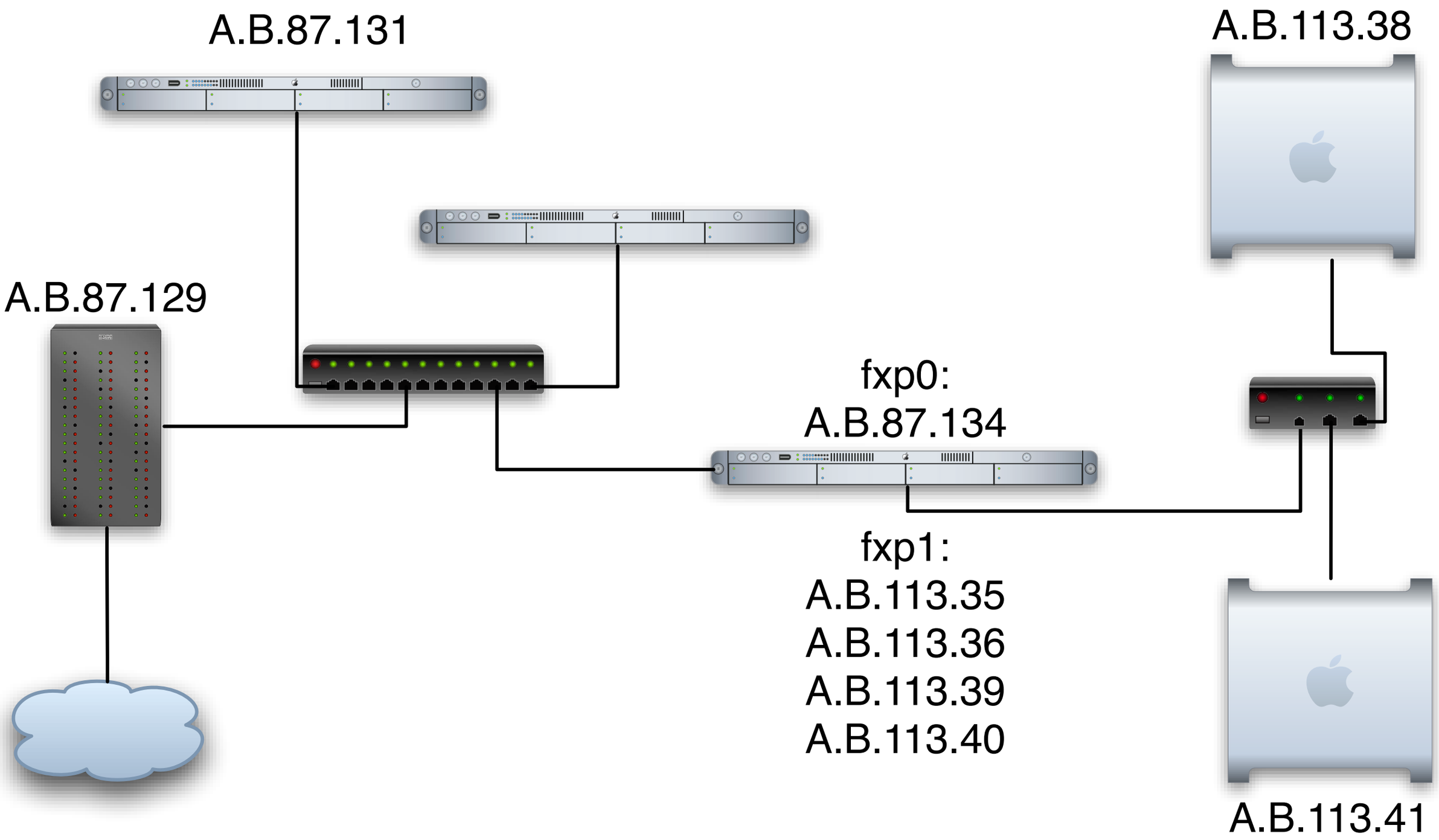
Our example Net



Terminal – RedTeam@RWTH

```
md@beebop:~ $ sudo honeyd -d -P -i fxp1
Honeyd V1.0 Copyright (c) 2002-2004 Niels Provos
honeyd[98979]: started with -d -P -i fxp1
Warning: Impossible SI range in Class fingerprint "IBM OS/400 V4R2M0"
Warning: Impossible SI range in Class fingerprint "Microsoft Windows NT 4.0 SP3"
honeyd[98979]: listening promiscuously on fxp1: (arp or ip proto 47 or (udp and src port 67
and dst port 68) or (ip )) and not ether src 00:10:83:fc:79:27
honeyd[98979]: switching to polling mode
honeyd[98979]: Demoting process privileges to uid 32767, gid 32767
honeyd[98979]: Connection to closed port: udp (A.B.113.38:53 - A.B.113.43:57823)
honeyd[98979]: Connection to closed port: udp (A.B.113.38:53 - A.B.113.43:58942)
honeyd[98979]: Connection to closed port: udp (A.B.113.38:3431 - 205.234.174.117:53)
honeyd[98979]: Connection to closed port: udp (A.B.113.38:47875 - 205.234.174.117:53)
honeyd[98979]: Connection to closed port: udp (A.B.113.38:24865 - 205.234.174.117:53)
honeyd[98979]: Connection to closed port: udp (A.B.113.38:53 - A.B.113.43:59153)
honeyd[98979]: Connection to closed port: udp (A.B.113.38:53 - A.B.113.43:59652)
honeyd[98979]: Connection to closed port: udp (A.B.113.38:53 - A.B.113.43:63083)
honeyd[98979]: Connection to closed port: udp (A.B.113.38:12757 - 205.234.174.117:53)
honeyd[98979]: Connection to closed port: udp (A.B.113.38:53 - A.B.113.43:58305)
honeyd[98979]: Connection to closed port: udp (A.B.113.38:42231 - 202.77.241.200:53)
honeyd[98979]: Connection to closed port: udp (A.B.113.38:53 - A.B.113.43:64039)
honeyd[98979]: Connection to closed port: udp (A.B.113.38:37780 - 63.149.146.4:53)
honeyd[98979]: Connection to closed port: udp (A.B.113.38:47488 - 205.234.145.230:53)
honeyd[98979]: Connection to closed port: udp (A.B.113.38:53 - A.B.113.43:61226)
honeyd[98979]: Connection to closed port: udp (A.B.113.38:53 - A.B.113.43:51982)
```

Our example Net



Terminal – RedTeam@RWTH

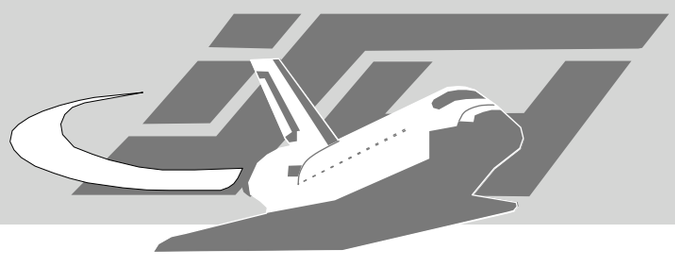
```
honeyd[98979]: Connection to closed port: udp (A.B.113.38:62570 - 192.55.83.30:53)
honeyd[98979]: Connection to closed port: udp (A.B.113.38:43552 - 192.26.92.30:53)
honeyd[98979]: Connection to closed port: udp (A.B.113.38:32276 - 192.12.94.30:53)
honeyd[98979]: Connection to closed port: udp (A.B.113.38:29443 - 192.42.93.30:53)
honeyd[98979]: Connection to closed port: udp (A.B.113.38:60608 - 192.33.14.30:53)
honeyd[98979]: Connection to closed port: udp (A.B.113.38:43828 - 192.35.51.30:53)
```

Terminal – RedTeam@RWTH

```
md@beebop:~ $ sudo honeyd -d -P -i fxp1 A.B.113.128/25
Honeyd V1.0 Copyright (c) 2002-2004 Niels Provos
honeyd[99900]: started with -d -P -i fxp1 A.B.113.128/25
Warning: Impossible SI range in Class fingerprint "IBM OS/400 V4R2M0"
Warning: Impossible SI range in Class fingerprint "Microsoft Windows NT 4.0 SP3"
honeyd[99900]: listening promiscuously on fxp1: (arp or ip proto 47 or (udp and src
port 67 and dst port 68) or (ip and (net A.B.113.128/25))) and not ether src
00:10:83:fc:79:27
honeyd[99900]: switching to polling mode
honeyd[99900]: Demoting process privileges to uid 32767, gid 32767
```

unreliable!

Part 4:
using clever routing



- Generate a dedicated interface for your honeypotting
- bind honeyd to that interface
- route the traffic meant for the honeynet to that interface

Terminal – RedTeam@RWTH

```
md@beebop:~ $ sudo ifconfig disc0 create

md@beebop:~ $ sudo ifconfig disc0 172.31.255.255 netmask 255.255.255.255

md@beebop:~ $ sudo honeyd -d -P -i disc0
Honeyd V1.0 Copyright (c) 2002-2004 Niels Provos
honeyd[2220]: started with -d -P -i disc0
honeyd[2220]: listening on disc0: ip
honeyd[2220]: switching to polling mode
honeyd[2220]: Demoting process privileges to uid 32767, gid 32767
```

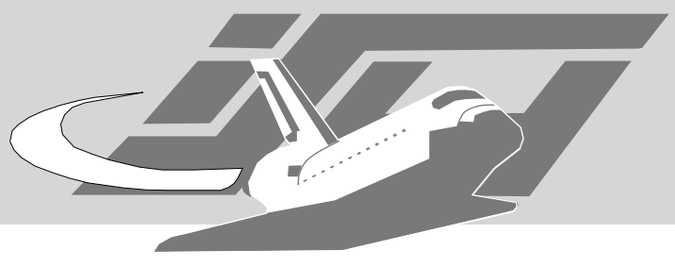
Terminal – RedTeam@RWTH

```
md@beebop:~ $ sudo route add -net A.B.113.128/27 -interface disc0
add net A.B.113.128: gateway disc0
```

Terminal – RedTeam@RWTH

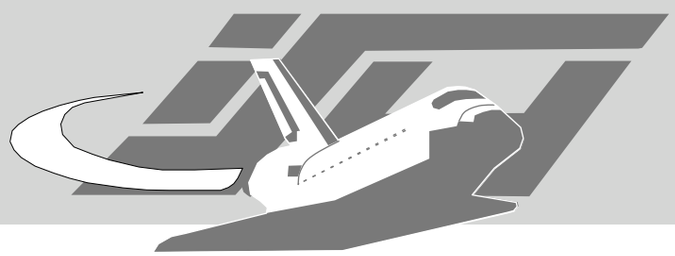
```
honeyd[2220]: Connection request: tcp (84.72.117.102:1850 - A.B.113.137:445)
honeyd[2220]: Connection established: tcp (84.72.117.102:1850 - A.B.113.137:445)
honeyd[2220]: Connection request: tcp (84.72.117.102:1855 - A.B.113.137:445)
honeyd[2220]: Connection closed: tcp (84.72.117.102:1850 - A.B.113.137:445)
honeyd[2220]: Connection established: tcp (84.72.117.102:1855 - A.B.113.137:445)
honeyd[2220]: Connection request: tcp (213.117.190.76:2756 - A.B.113.133:445)
honeyd[2220]: Connection established: tcp (213.117.190.76:2756 - A.B.113.133:445)
honeyd[2220]: Connection request: tcp (213.117.190.76:2782 - A.B.113.133:139)
honeyd[2220]: Connection closed: tcp (213.117.190.76:2756 - A.B.113.133:445)
honeyd[2220]: Connection established: tcp (213.117.190.76:2782 - A.B.113.133:139)
honeyd[2220]: Connection closed: tcp (213.117.190.76:2782 - A.B.113.133:139)
```

Part 5:
using honeyd 1.0 features



Honeyd 1.0 features

- pro
  - generate arp replies
  - request addresses via DHCP
  - don’t break your network
- contra
  - seemingly little used & documented



arp by honeyd

- set <template> ethernet "<vendor>"
- set <template> ethernet "<mac address>"
- bind <ip> <template>

Terminal – RedTeam@RWTH

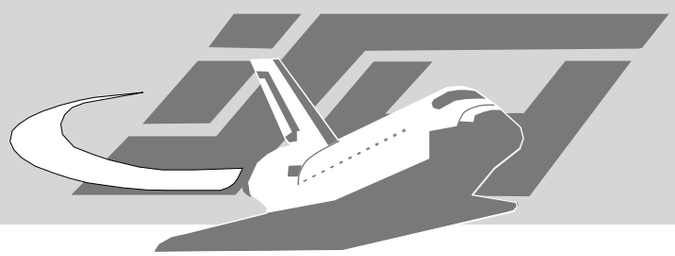
```
create default
set default personality "Microsoft Windows XP Home Edition"
[...]
### Standard Windows 2000 computer
create win2k
set win2k ethernet "intel"
bind 137.226.59.100 win2k
bind 137.226.59.101 win2k

### Linux Suse 8.0 template
create suse80
set suse80 personality "Linux 2.4.7 (X86)"
[...]
set suse80 ethernet "3com"
bind 137.226.59.103 suse80
bind 137.226.59.104 suse80

### Suse7.0 computer
create suse70
set suse70 personality "Linux 2.2.12 - 2.2.19"
[...]
set suse70 ethernet "eltec"
bind 137.226.59.105 suse70
bind 137.226.59.106 suse70
```

Terminal – RedTeam@RWTH

```
dornseif@houston:/opt/partimage/md/honeyd_kit-1.0c-a$ sudo ./start-honeyd.sh
+ ./honeyd -d -f honeyd.conf -p nmap.prints -x xprobe2.conf -a nmap.assoc -0 pf.os -l /var/
log/honeyd 137.226.59.100-137.226.59.120
Honeyd V1.0c Copyright (c) 2002-2004 Niels Provos
[warn] epoll_create: Function not implemented
honeyd[30163]: started with -d -f honeyd.conf -p nmap.prints -x xprobe2.conf -a nmap.assoc -0
pf.os -l /var/log/honeyd 137.226.59.100-137.226.59.120
honeyd[30163]: listening promiscuously on eth0: (arp or ip proto 47 or (udp and src port 67
and dst port 68) or (ip and (dst net 137.226.59.100/30 or dst net 137.226.59.104/29 or dst
net 137.226.59.112/29 or dst net 137.226.59.120/32))) and not ether src 00:04:76:0c:0d:e5
honeyd[30163]: Demoting process privileges to uid 65534, gid 65534
honeyd[30163]: honeyd_logstart: fopen("/var/log/honeyd")
honeyd[30163]: Killing attempted connection: tcp (137.226.59.38:64039 - 137.226.59.104:716)
honeyd[30163]: arp_send: who-has 137.226.59.38 tell 137.226.59.104
honeyd[30163]: Killing attempted connection: tcp (137.226.59.38:64040 - 137.226.59.104:1377)
honeyd[30163]: Killing attempted connection: tcp (137.226.59.38:64041 - 137.226.59.104:3264)
honeyd[30163]: Killing attempted connection: tcp (137.226.59.38:64042 - 137.226.59.104:2017)
honeyd[30163]: Killing attempted connection: tcp (137.226.59.38:64043 - 137.226.59.104:312)
honeyd[30163]: Killing attempted connection: tcp (137.226.59.38:64045 - 137.226.59.104:2627)
honeyd[30163]: Killing attempted connection: tcp (137.226.59.38:64046 - 137.226.59.104:694)
```

Terminal – RedTeam@RWTH

```
19:25:01.588323 arp who-has 137.226.59.100 tell 137.226.59.38
[...]
19:25:01.588974 arp who-has 137.226.59.120 tell 137.226.59.38
19:25:01.589011 arp reply 137.226.59.100 is-at 00:a0:c9:d9:2e:47
19:25:01.589006 arp who-has 137.226.59.121 tell 137.226.59.38
19:25:01.589036 arp who-has 137.226.59.122 tell 137.226.59.38
19:25:01.589077 arp who-has 137.226.59.123 tell 137.226.59.38
19:25:01.589154 arp reply 137.226.59.101 is-at 00:a0:c9:63:f5:6d
19:25:01.589230 arp who-has 137.226.59.124 tell 137.226.59.38
19:25:01.589288 arp reply 137.226.59.103 is-at 00:d0:d8:ca:53:ed
19:25:01.589315 arp who-has 137.226.59.125 tell 137.226.59.38
19:25:01.589411 arp reply 137.226.59.105 is-at 00:00:5b:ef:fe:50
19:25:01.589482 arp reply 137.226.59.104 is-at 00:d0:d8:5e:e2:b6
19:25:01.589585 arp reply 137.226.59.106 is-at 00:00:5b:18:c4:f0
19:25:01.590256 arp who-has 137.226.59.38 (ff:ff:ff:ff:ff:ff) tell 137.226.59.100
19:25:01.590362 arp reply 137.226.59.38 is-at 00:0d:93:2f:82:64
19:25:01.897099 arp who-has 137.226.59.125 tell 137.226.59.38
19:25:01.897148 arp who-has 137.226.59.124 tell 137.226.59.38
19:25:01.897185 arp who-has 137.226.59.122 tell 137.226.59.38
```



dhcp by honeyd

- does not work when honeyd is running on the dhcp server
- needs the -i parameter
- do not use the same ethernet address twice

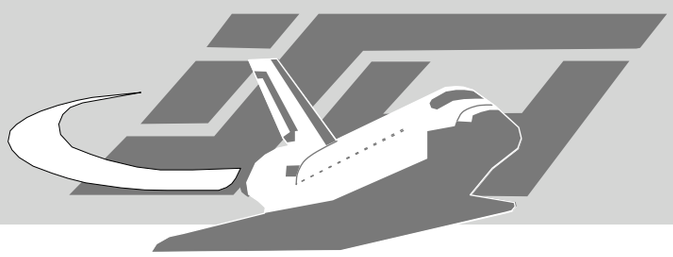
Terminal – RedTeam@RWTH

```
[...]
# DHCP honeypots
dhcp win2k on xl0 ethernet "3com"
dhcp win2k on xl0 ethernet "3com"
dhcp win2k on xl0 ethernet "3com"
dhcp suse80 on xl0 ethernet "intel"
dhcp suse70 on xl0 ethernet "intel"
```

Terminal – RedTeam@RWTH

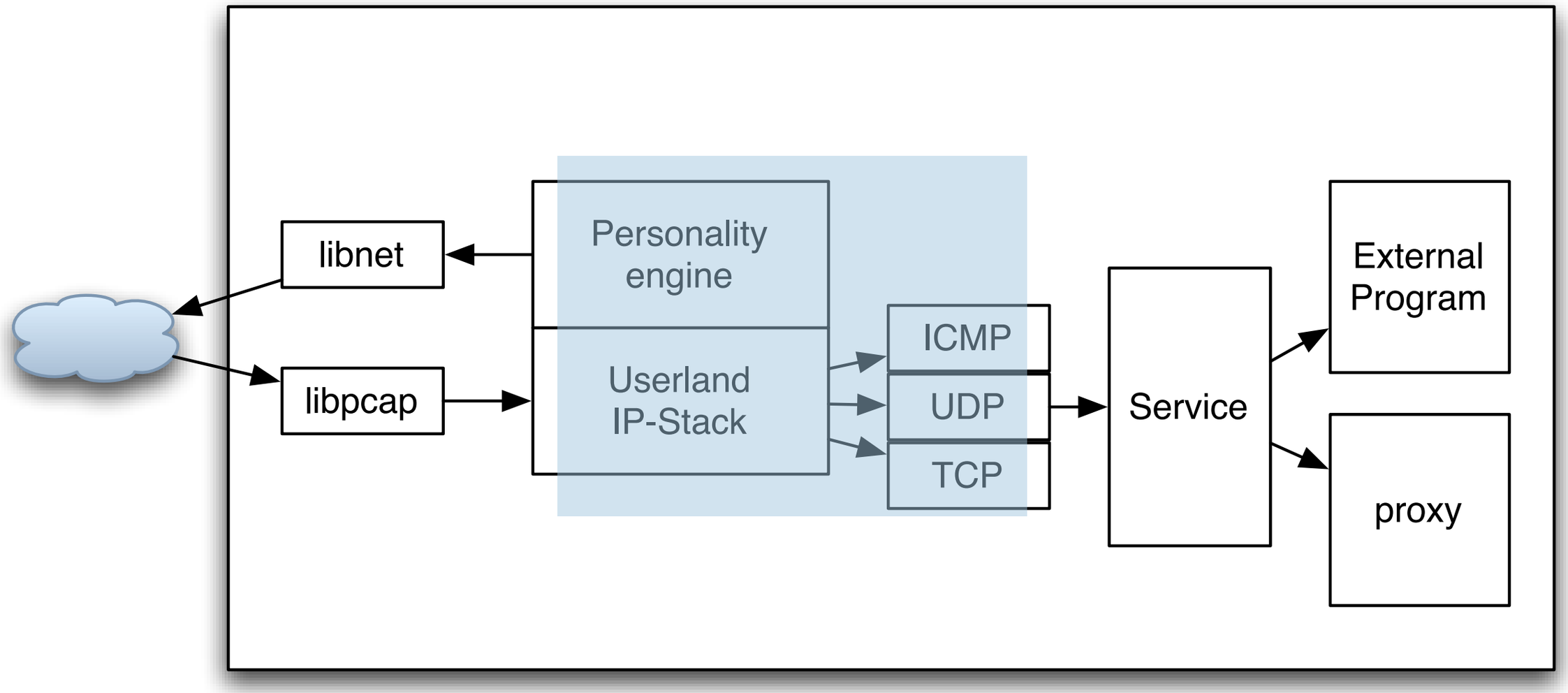
```
lying-bastard# sh start-honeyd.sh
+ honeyd -d -P -i xl0 -f honeyd.conf -p nmap.prints -x xprobe2.conf -a nmap.assoc -0 pf.os -l /var/
log/honeyd 137.226.59.12-137.226.59.126
Honeyd V1.0 Copyright (c) 2002-2004 Niels Provos
honeyd[45008]: started with -d -P -i xl0 -f honeyd.conf -p nmap.prints -x xprobe2.conf -a
nmap.assoc -0 pf.os -l /var/log/honeyd 137.226.59.12-137.226.59.126
Warning: Impossible SI range in Class fingerprint "IBM OS/400 V4R2M0"
Warning: Impossible SI range in Class fingerprint "Microsoft Windows NT 4.0 SP3"
honeyd[45008]: listening promiscuously on xl0: (arp or ip proto 47 or (udp and src port 67 and dst
port 68) or (ip and (dst net 137.226.59.12/30 or dst net 137.226.59.16/28 or dst net
137.226.59.32/27 or dst net 137.226.59.64/27 or dst net 137.226.59.96/28 or dst net
137.226.59.112/29 or dst net 137.226.59.120/30 or dst net 137.226.59.124/31 or dst net
137.226.59.126/32))) and not ether src 00:60:08:6e:34:3a
honeyd[45008]: switching to polling mode
honeyd[45008]: HTTP server listening on port 80
honeyd[45008]: HTTP server root at /usr/local/share/honeyd/webserver/htdocs
honeyd[45008]: [xl0] trying DHCP
[...]
honeyd[45008]: Demoting process privileges to uid 32767, gid 32767
honeyd[45008]: honeyd_logstart: fopen("/var/log/honeyd")
honeyd[45008]: [xl0] got DHCP offer: 137.226.59.103
honeyd[45008]: Updating ARP binding: 3c:00:00:8d:d7:95 -> 137.226.59.103
[...]
honeyd[45008]: arp reply 137.226.59.103 is-at 3c:00:00:8d:d7:95
[...]
honeyd[45008]: Sending ICMP Echo Reply: 137.226.59.103 -> 137.226.59.9
[...]
honeyd[45008]: Killing unknown connection: tcp (137.226.59.91:35166 - 137.226.59.103:80)
honeyd[45008]: arp_send: who-has 137.226.59.91 tell 137.226.59.103
honeyd[45008]: arp_recv_cb: 137.226.59.9 at 00:04:76:0c:0d:e5
_Segmentation fault
```

Honeyd configuration



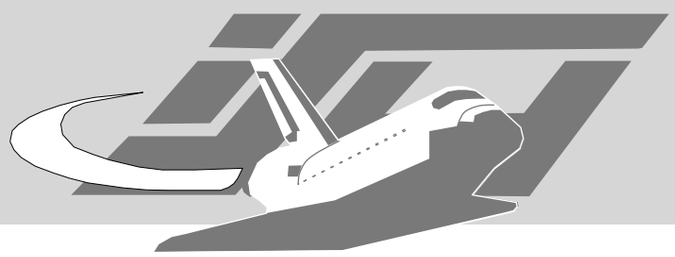
- Somewhat underdocumented
- man honeyd

What IP-Stacks to emulate?



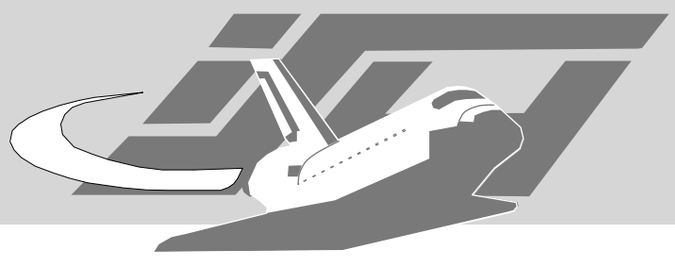
Terminal – RedTeam@RWTH

```
# Set default behavior
create default
add default tcp port 139 open
add default tcp port 137 open
add default udp port 137 open
add default udp port 135 open
set default personality "Microsoft Windows XP Home Edition"
set default default tcp action reset
set default default udp action block
set default default icmp action open
set default uptime 234217
set default droprate in 5
set default uid 17 gid 17
set default maxfds 127
set default ethernet "apple"

create chaos
set chaos personality random
set default uptime 666
# see ethernet.c
set chaos ethernet "raytheon company"
```



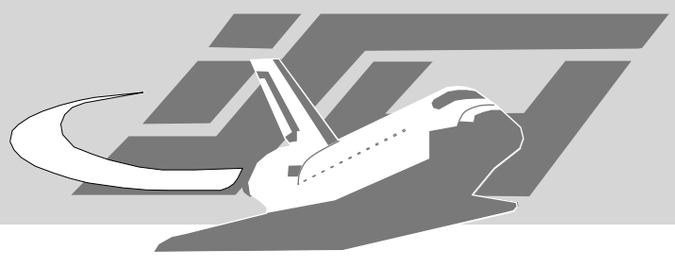
actions

- block
- open
- reset
- [internal] <cmd-string>
- proxy <ipaddr>:<port>
- tarpit



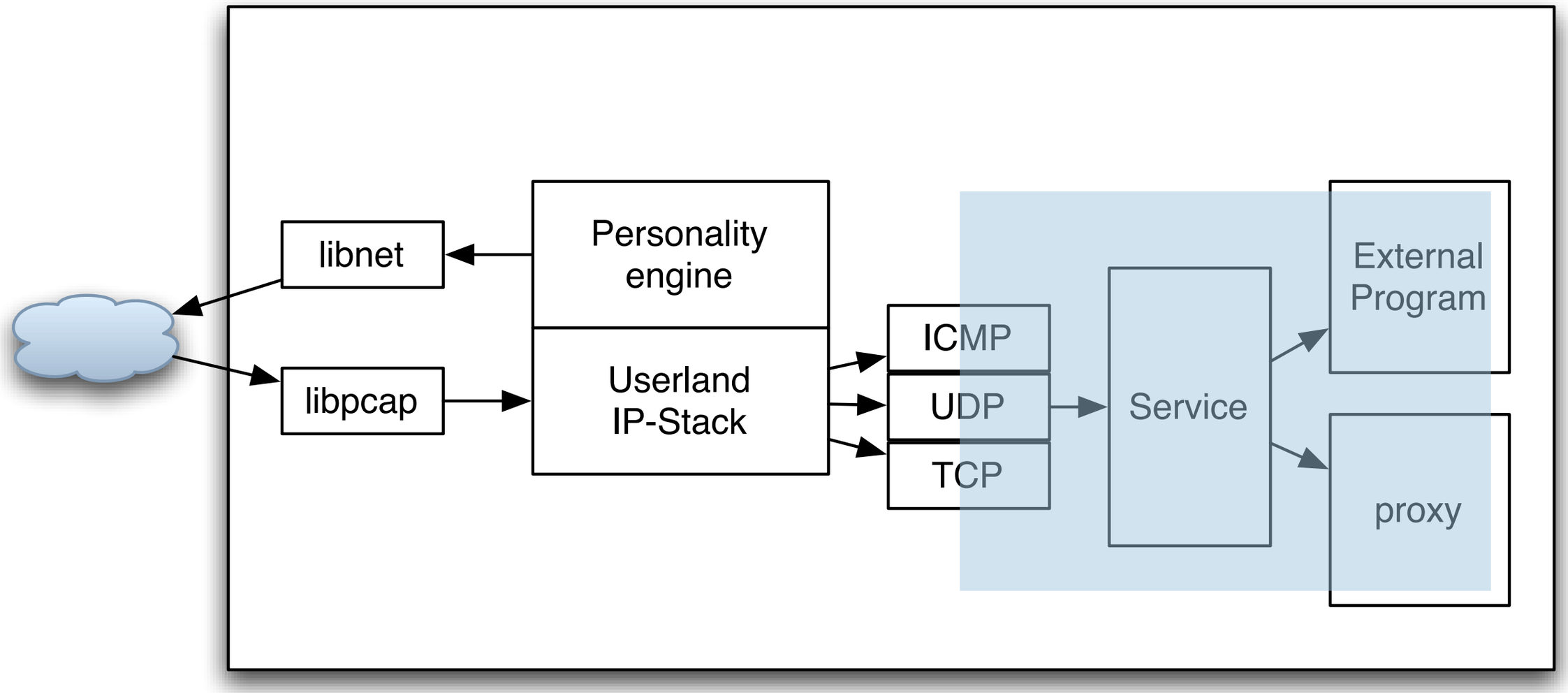
- set default uptime 234217
- set default droprate in 5
- set default uid 17 gid 17
- set default maxfds 127
- set default ethernet "apple"

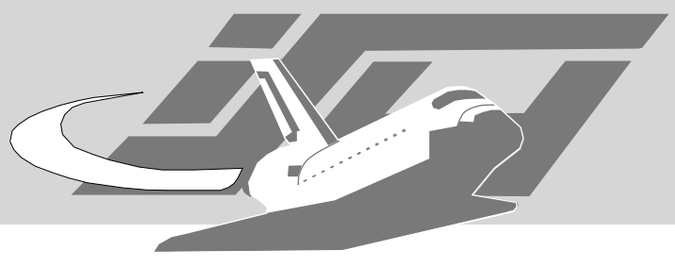
What gets to the Stack?



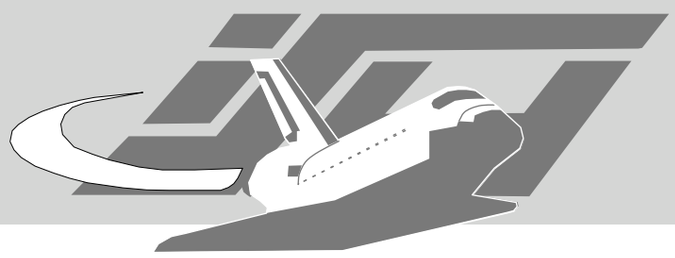
- bind <ip-address> <template-name>
- bind <ip-address> <interface-name>
- dhcp <template-name> on <interface-name> [ethernet <vendor>]
- clone <template-name> <template-name>

What Services to emulate?



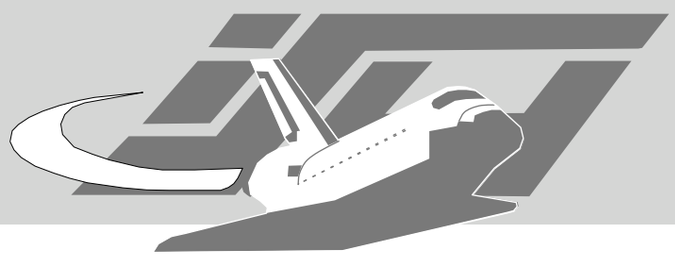


- Scripts
- Plugins
- Proxying
- Overloading / “Subsystem Virtulization”



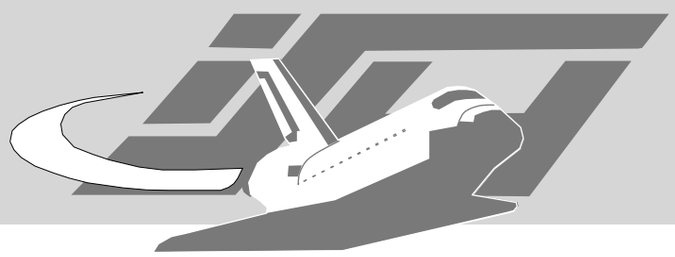
Scripts

- add template tcp port 80 "<scriptpath&name>"
- sources for scripts:
  - http://www.honeyd.org/contrib.php
  - http://www.honeynet.org.br/tools/
  - http://sourceforge.net/projects/iisemul8/
  - http://www.citi.umich.edu/u/provos/honeyd/honeyd_kit-1.0c-a.tgz



Plugins

- add template tcp port 23 internal "<mymodule>"
- no path
- no extension
- more on that later



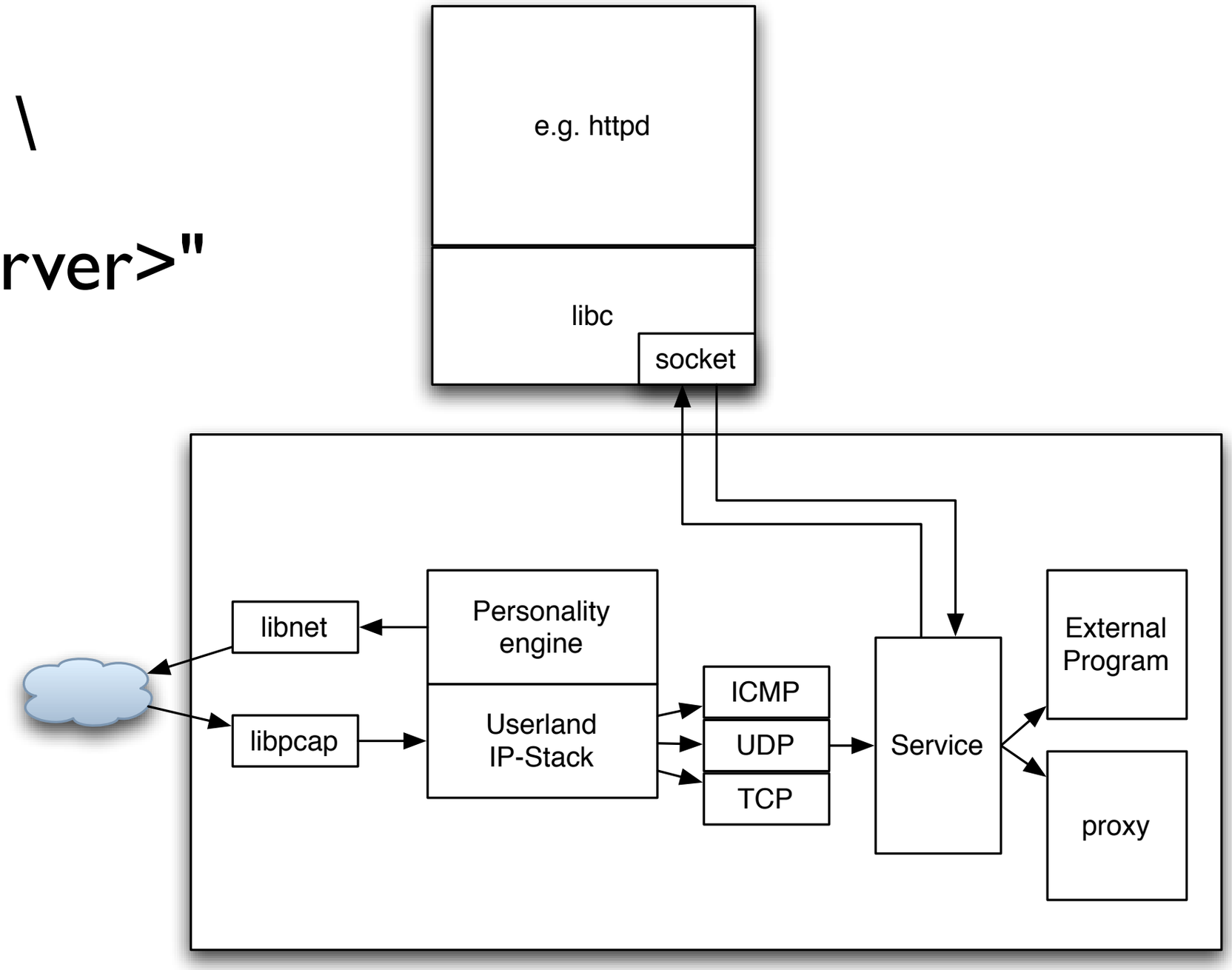
Proxying

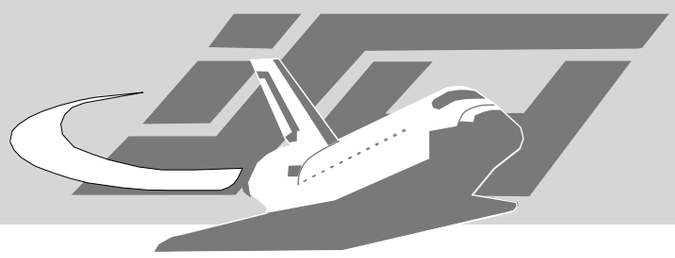
- add template tcp port 23 proxy $ipsrc:23
- add template udp port 53 proxy a.b.c.d.e:53

Subsystem Virtulization

- set <template> \

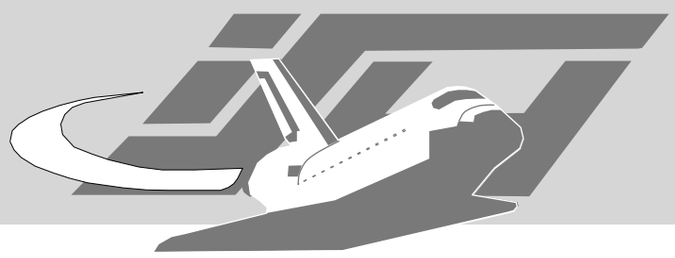
  subsystem "<server>"





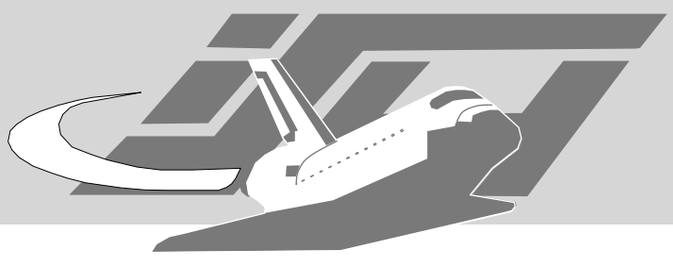
Environment & Co

- $ipsrc, $ipdst, $sport $dport
- HONEYD_IP_SRC
- HONEYD_IP_DST
- HONEYD_DST_PORT
- HONEYD_SRC_PORT
- HONEYD_PERSONALITY
- HONEYD_REMOTE_OS



Dynamic Templates

- Assignments based on conditions
  - what is that good for?

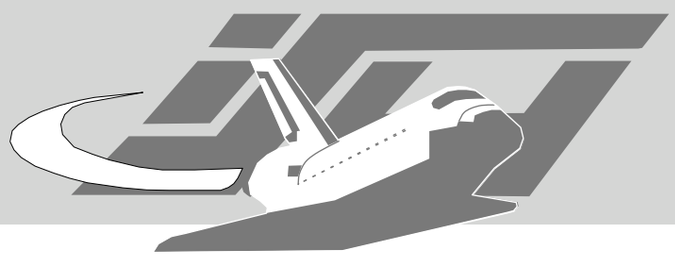


Terminal – RedTeam@RWTH

```
### Dynamic honeypot
dynamic magichost
add magichost use mac if source os = "windows"
add magichost use suse70 if source ip = 192.168.1.0/28
add magichost use router if time between 12:00am - 5:00am
add magichost otherwise use default
bind a.b.1.100 magichost
bind a.b.1.101 magichost
bind a.b.1.102 magichost
```

Simulating Services

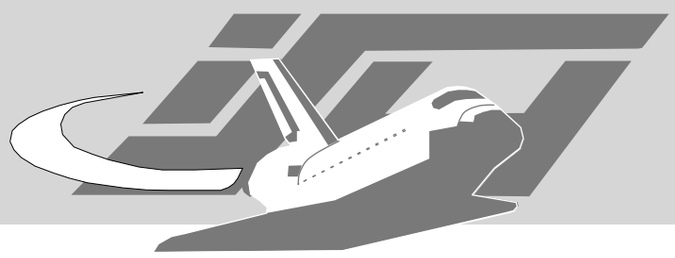
iis emulator



- “The goal of this project is to create a functioning web server which is indistinguishable from Microsoft's IIS product at a topical level. This server can be run standalone, through inetd, or as a module of the ‘honeyd’ project.”

- http://sourceforge.net/projects/iisemul8

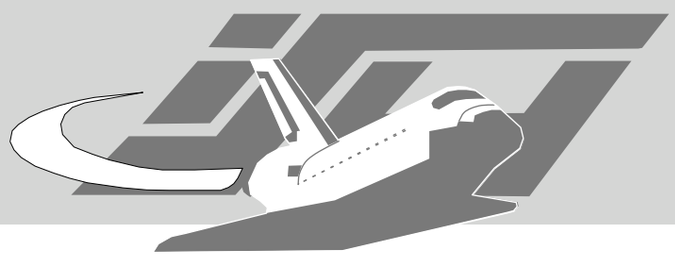
Honeybee



- Diploma Thesis done at our Lab
- Semi-Automatically generates emulators for arbitary
  - currently supports SMTP, FTP and POP3
- http://thomas-apel.de/honeybee/

Macintosh HD

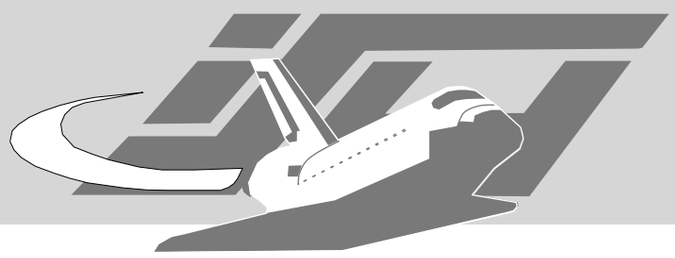
Demo



Terminal – RedTeam@RWTH

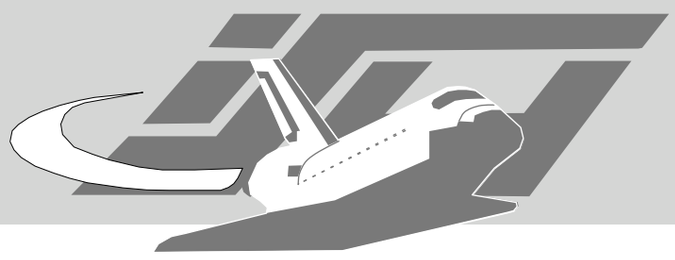
```
# honeyd.conf
create default
set default personality random
#add default tcp port 80 "scripts/iis5.net/main.pl"
set default default tcp action open
set default default udp action open
set default default icmp action open

create foobar
set foobar ethernet "3com"

bind 10.17.23.151 foobar
# - or -
dhcp foobar on eth0 ethernet "3com"
```

Lance Spitzner's Honeyd Linux Kit



honeyd toolkit

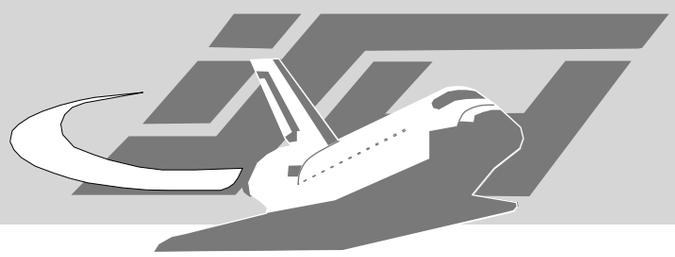
- Precompiled Linuix Toolkit
  - Also very nice if you don’t run Linux
- http://www.tracking-hackers.com/solutions/honeyd/
- http://www.citi.umich.edu/u/provos/honeyd/honeyd_kit-1.0c-a.tgz



Things you have to change

- start-arpd.sh
  - better go without arpd
- start-honeyd.sh
- honeyd.conf

HOACD



HOACD

- The brazilians Honeynet bootable honeyd CD
- not unlike the Honeywall CD
- http://www.honeynet.org.br/tools/

Gatering Results

Finder File Edit View Go Window Help demo

Honeyd Administration Interface

http://10.17.23.13/index.py

pdcn DLF Python Selfhtml B! WB! del.icio.us MarsEdit tickets CuRM LuFG debug lufgi4 CIPHER 2005... Stormcrow

Honeyd Administration Interface

Uptime: 0 days 00:08:11

Honeyd Administration

Main
Configuration

Honeyd Search

Google

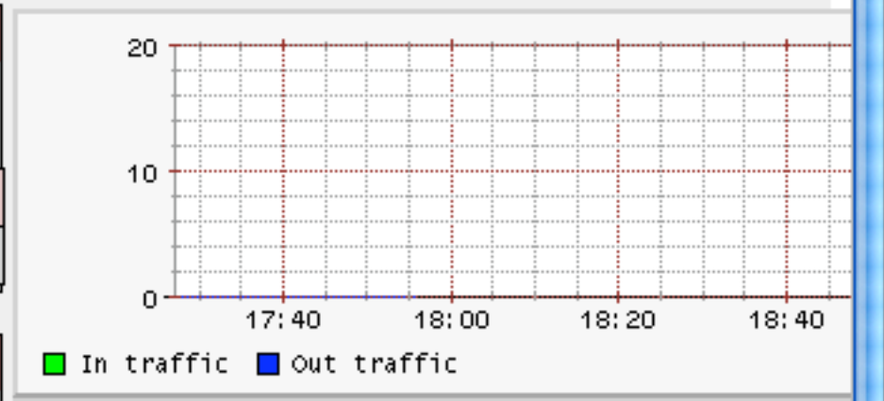
Web Honeyd.org Search

Welcome to the Honeyd Administration Interface.You are visitor 3.

Interface Information

This table shows the interface that Honeyd has been configured to listen to.

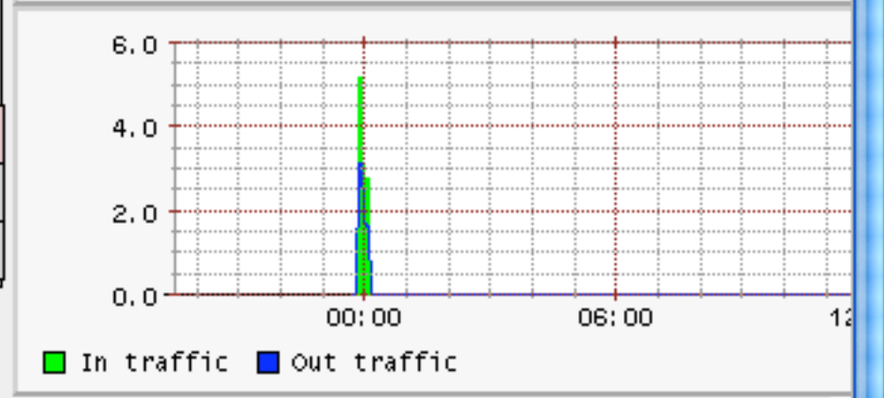
Name	Address	MTU	Link Address
eth0	10.17.23.13	1500	00:0c:29:06:23:db



Honeyd Statistics

This table shows current statistics collected by Honeyd.

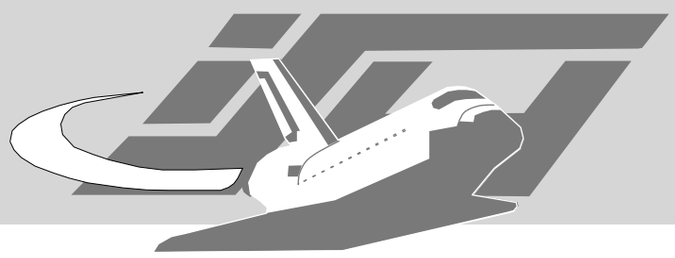
Variable	Minute	Hour	Day
Input Bytes	300.92 B/s	11.79 B/s	0.00 B/s
Output Bytes	86.40 B/s	0.64 B/s	0.00 B/s



There are currently no active TCP connections.

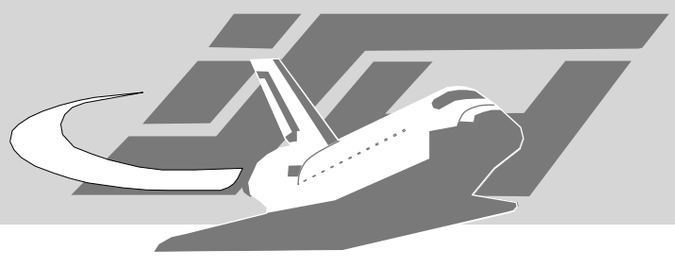
There are currently no active UDP connections.

Visit www.honeyd.org for more information.



The honeyd collector interface

- Added in honeyd 1.0
- undocumented
- no public collector implementation exists
  - until today at least
- -c host:port:name:pass



The honeyd collector interface

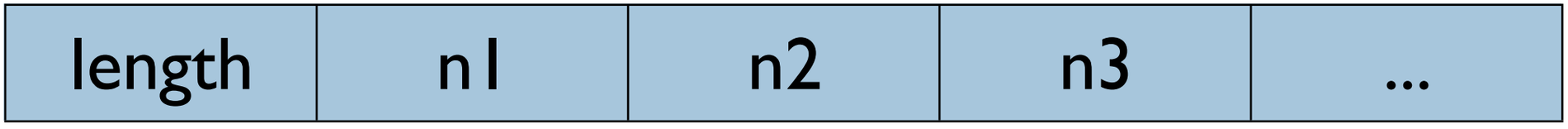
- Take logging information
- pack it smartly into udp packets
- ‘sign’ it by a HMAC
- slap an username on it
- send it to a central server

honeyd ‘tags’

tag_id	length	data

- tag_id identifies the following data
  - is contenxt dependant
- length is an encoded integer
  - length has variable length : 1-9 bytes
- data is length bytes long
- tag size is 1 + len(length) + length

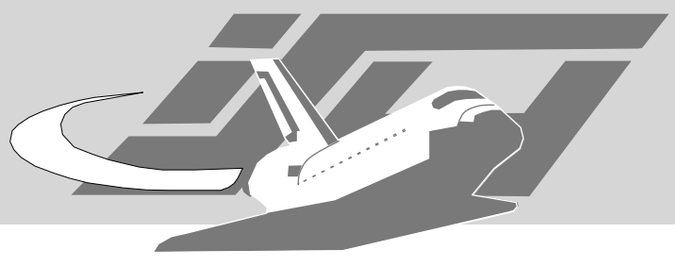
encoded integers



- encoding works with nibbles (4bits) not with bytes (8 bits)
- the length nibble encodes the number of following nibbles -1
  - max 16 nibbles
  - max 64 bit

Terminal – RedTeam@RWTH

```
def decode_int(self, data):

    nibbles = ((ord(data[0]) & 0xf0) >> 4) + 1
    intdata = data[0:(nibbles//2)+1]
    number = []
    offset = 1
    while offset <= nibbles:
      if offset & 1:
        number.append(ord(data[offset//2]) & 0x0f)
      else:
        number.append((ord(data[offset//2]) >> 4) & 0x0f)
      offset += 1;
  number.reverse()
  ret = 0
  for x in number:
    ret = ret << 4
    ret += x
  return ret, (nibbles//2) + 1
```



SIG tags

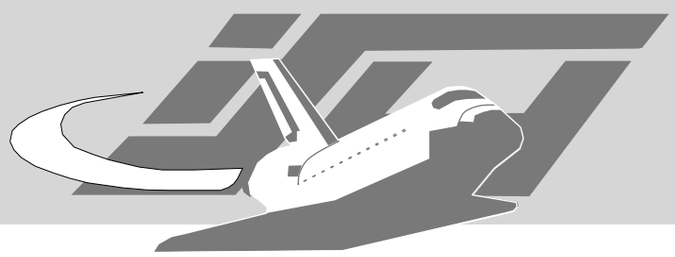
tag_id	length	data

- The outermost layer of honeyd communication are SIG tags
- SIG_NAME, SIG_DIGEST, SIG_DATA, SIG_COMPRESSED_DATA, SIG_MAX
- Real-world UDP packets consist of SIG_NAME, SIG_DIGEST, SIG_COMPRESSED_DATA

- SIG_NAME tags are identified by the value 0

- SIG_NAME tags just contain the name you have given as the third value in the -c parameter to honeyd

```
00 02 6d 64 01 14 10 96   b3 ef cb 92 2e 40 f8 b6   ..md.........@..
```

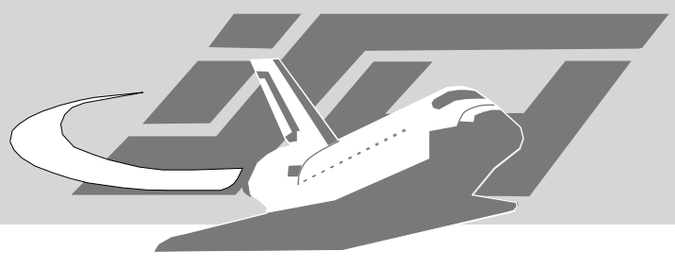


- SIG_DIGEST tags are identified by the value 1
- SIG_DIGEST tags contain a SHA-1 HMAC to enable the receiver to verify the authentic and integrity of the data.
- The HMAC is based on the ‘pass’ you have given as the fourth value in the -c parameter to honeyd

```
00 02 6d 64 01 14 10 96   b3 ef cb 92 2e 40 f8 b6   ..md.........@..
61 cd 56 b3 e1 11 56 d2   f1 dd 6e 03 2e c1 78 da   a.V...V...n...x.
62 60 92 be c0 c8 51 35   89 e7 92 83 6b ae 21 13   b`....Q5....k.!.
```

- SIG_COMPRESSED_DATA tags are identified by the value 3
- SIG_COMPRESSED_DATA tags ontain a bunch of zlib compressed measurement tags.

```
00 02 6d 64 01 14 10 96    b3 ef cb 92 2e 40 f8 b6    ..md.........@..
61 cd 56 b3 e1 11 56 d2    f1 dd 6e 03 2e c1 78 da    a.V...V...n...x.
62 60 92 be c0 c8 51 35    89 e7 92 83 6b ae 21 13    b`....Q5....k.!.
                                    ...
```

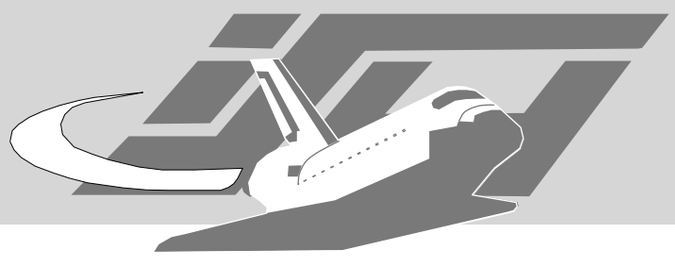


NAME	DIGEST	COMPRESSED_DATA

- COMPRESSED_DATA has to be decoded:

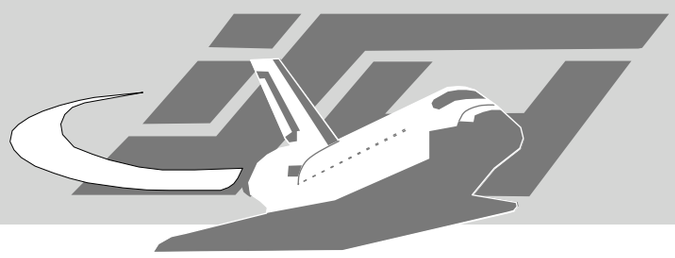
```
class TagSig(Tag):
  def parsetyp(self):
    self.typname = SIGs[self.typ]
  if self.typ == SIG_COMPRESSED_DATA:
    d = zlib.decompressobj()
    self.data = d.decompress(self.data)
    data  = self.data
    while data:
      t = TagMeasurement()
      data = t.parse(data)
      self.subtags.append(t)
```

- Then we have a bunch of measurement tags



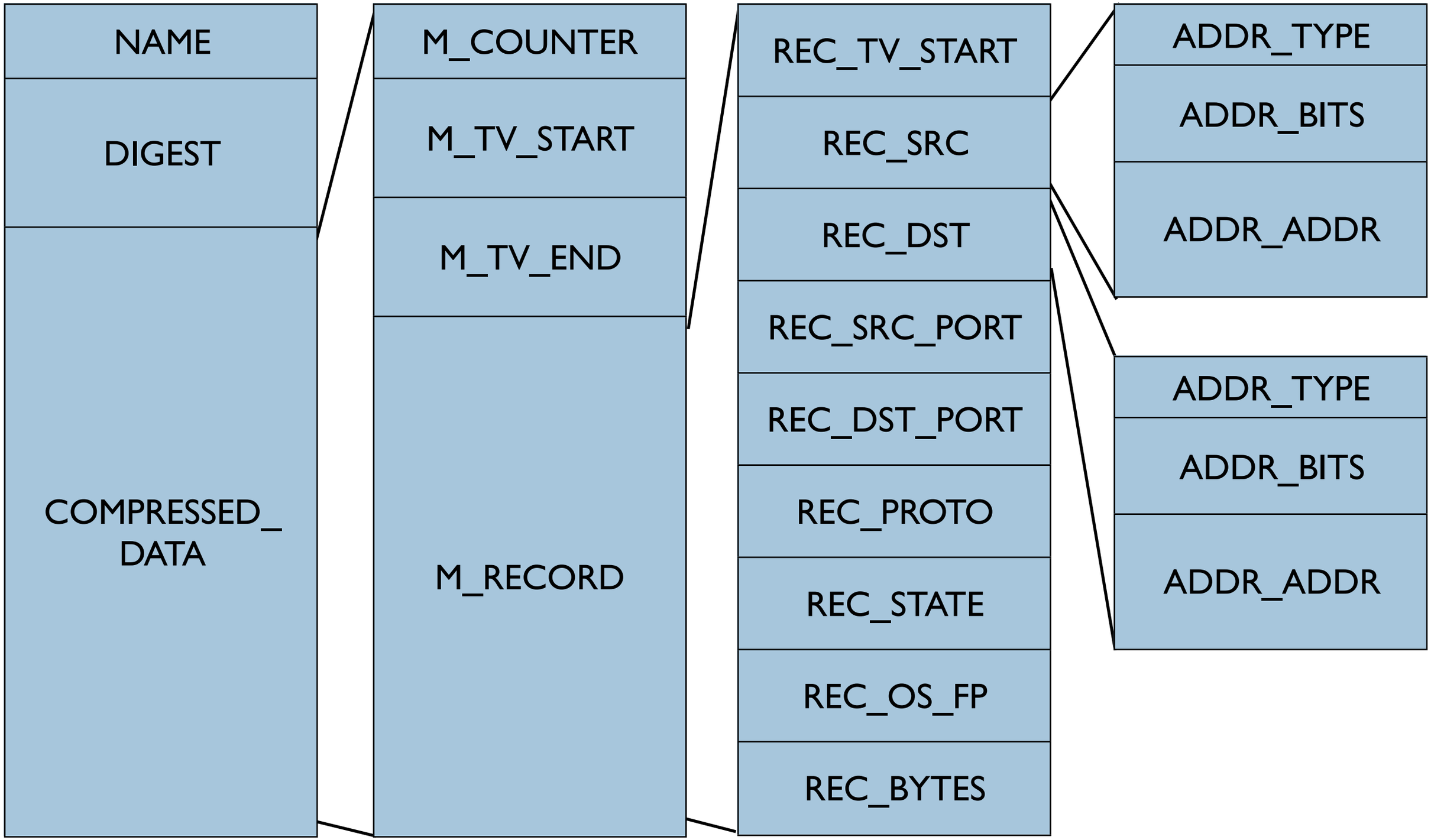
- M_COUNTER, M_TV_START, M_TV_END, M_RECORD, M_MAX
- M_TV_START and M_TV_END are encoded time values describing the timespan during which the measurments where taken.
- Time is encoded as 2 integers:

```
def decode_timeval(self, data):
  sec, bytes = self.decode_int(data)
  usec, bytes = self.decode_int(data[bytes:])
  return (sec, usec)
```



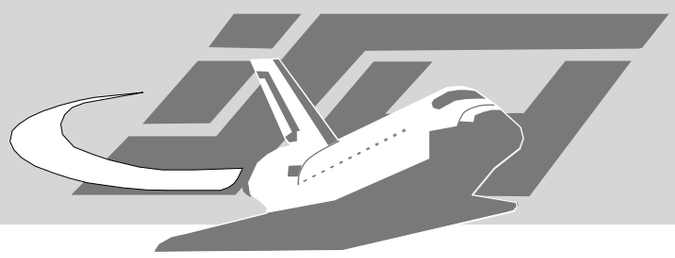
- M_RECORD contains several record tags which in turn contain the real meat.
- Typical record tags: REC_SRC, REC_DST, REC_SRC_PORT, REC_DST_PORT, REC_PROTO, REC_STATE, REC_OS_FP, REC_BYTES
- REC_ SRC and REC_DST contain ADDR tags
  - which contain IP or ethernet adresses, spread over 3 tags: ADDR_TYPE, ADDR_BITS, ADDR_ADDR

all together now



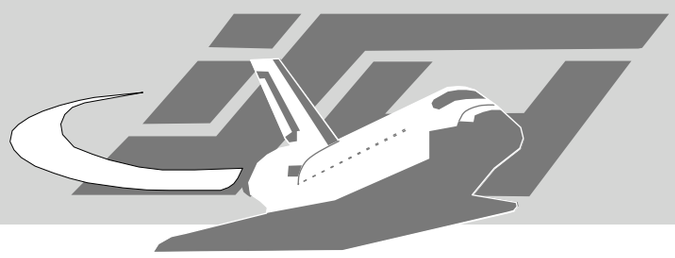
Terminal – RedTeam@RWTH

```
$ sudo honeyd -d -P -l ./honeyd-packet.log -s ./honeyd-service.log
-i fxp0 -c lufgi4.informatik.rwth-aachen.de:5711:lufg:blafasel
Honeyd V1.0 Copyright (c) 2002-2004 Niels Provos
honeyd[87157]: started with -d -P -l ./honeyd-packet.log -s ./honeyd-service.log -i fxp0
-c lufgi4.informatik.rwth-aachen.de:5711:lufg:blafasel
Warning: Impossible SI range in Class fingerprint "IBM OS/400 V4R2M0"
Warning: Impossible SI range in Class fingerprint "Microsoft Windows NT 4.0 SP3"
honeyd[87157]: listening promiscuously on fxp0: (arp or ip proto 47 or (udp and src port
67 and dst port 68) or (ip )) and not ether src 00:d0:b7:c6:0f:a7
honeyd[87157]: switching to polling mode
honeyd[87157]: Demoting process privileges to uid 32767, gid 32767
honeyd[87157]: honeyd_logstart: fopen("./honeyd-packet.log")
honeyd[87157]: honeyd_logstart: fopen("./honeyd-service.log")
honeyd[87157]: Creating new stats buffer for (222.76.165.112:3558 - A.B.113.74:1434)
honeyd[87157]: Connection to closed port: udp (222.76.165.112:3558 - A.B.113.74:1434)
honeyd[87157]: Creating new stats buffer for (213.255.85.95:3248 - A.B.113.184:139)
honeyd[87157]: Connection request: tcp (213.255.85.95:3248 - A.B.113.184:139)
```



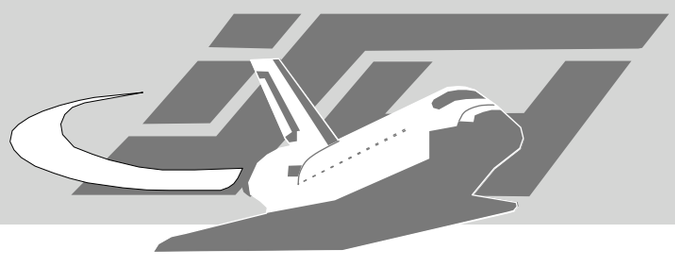
mustard - the free honeyd collector

- ...
- http://svn.23.nu/svn/repos/mustard/



Conclusion

- honeyd is an extremly powerfull tool
- it is somewhat underdocumented, you have to experiment with it
- you probably have to create your own code arround honeyd



Schedule

	Monday	**Tuesday**
9:00 10:30	Intro	forensics
10:45 12:30	Gen II/III Honeynets	botnets
14:00 16:00	honeyd distributed honeynets	attacks
16:15 18:00	nwcollect nephentis	leurre honeyfarms outlook