TODO

Leon Kaiser

2011-05-31

Contents

1 Tasks Organized by Priority

1.1 High Priority

- Support clone connections via `TOR`.
  - `TOR` might actually already be supported by `PROXY`.
- File flooding with *optional* adjustments for nick-length and latency.
  - The reason for the 'optional' bit is because one won't necessarily need a nick-length adjustment if the file being flooded is not an `ASCII` file (in fact, it will just look weird.)
- Permit (bot)net to target multiple networks (be able to send separate commands to an individual network's bots.
- Implement `STUPID`-like `SSH` tunneling. (`SSH Tunnel Utilizing Python IRC Destroyer`.)
- `nickspam.pl`-style nick spamming (using `/NAMES` output.)

1.2 Medium Priority

- Add 'nickshuffle' with 'nickbase' option.
- `do_jupe()`:
  - Call on {de,re,}connection.
- muhstik's mechanism to do `MODE`s and `KICK`s sucks. It doesn't track any list of nicks to op, or nicks to `KICK`, but it tries to change with simple `MODE +o` and `KICK`. This needs to be recoded with penalty handling.
- Mass{`KNOCK`,`INVITE`,`TOPIC`}.
  - For mass`INVITE`s, allow a mask (*!*@*.* style) 'blacklist' of sorts, so that the clones don't `INVITE` honeypot bots, lorf.
- Add a 'stop connect' command ('pause' or w/e.)
  - It should toggle, obviously.
- Max bots for connect – maximum successful bots, not maximum connections.
  - Allow increase/decrease.

- Support `SSL/TLS` and `SASL IRC` connections[1][2][3]
- Fix the buffer overflow or what the fuck ever when scan mode is enabled.

1.3 Low Priority

- Inline documentation.
- `TOPIC` lock mode.
- `TOPIC` fight mode.
- `CTCP` responses.
- Random colored messages.
- Spam every person, (non-){op,ircop,voice}, etc. in the channel.
- {Mass,Single} reconnect (for evading `ident` bans.)
- Detect `MODE +g` notifications on `PRIVMSG`.
  - Probably should just issue a warning or something.
- Add *optional* spectator – bot that watches, and doesn't respond to 'all'.
- Modify bot behavior so that if a clone is the only 'user' in a channel, the clone will cycle *only* once. If deopped by ChanServ, etc. then give up.
  - Possibly include a configuration setting that determines if channel registration is possible on the network (`boolean`, on the off-chance that a network with NickServ doesn't have ChanServ, as it would otherwise be covered by `conf.dalnet`.)
- Does `echo` mimic `CTCP ACTION`s as well?
- Allow *optional* Cisco passwords (in format IP{:PW,}.)
- `do_jupe()`:
  - If ghosted, does the affected clone reconnect?

[1] Leach, P., Newman, C. *Using Digest Authentication as a SASL Mechanism*, `RFC 2831`, May 2000, (http://www.ietf.org/rfc/rfc2831.txt)

[2] Myers, J. *Simple Authentication and Security Layer (SASL)*, `RFC 2222`, October 1997. (http://www.ietf.org/rfc/rfc2222.txt)

[3] Newman, C. *Anonymous SASL Mechanism*, `RFC 2245`, November 1997. (http://www.ietf.org/rfc/rfc2245.txt)

1.4 Unknown Priority

- `do_jupe()`:
  - _2_ @ `MONITOR`
  - Handle overflow.
- `.select *&,`
- `.deaf`
- Write or find some sort of tool that can differentiate between `SOCKS4` and `SOCKS5` proxies.
- Should `static` be used more?

2 Relevant IRC/IRL logs:

2.1 Jmax

2.1.1 Jmax and madvirii

02:32:57 <@Jmax> maybe... add line numbers?
02:33:12 <+madvirii> yah, like but it would have to be a system
02:33:19 <@Jmax> or after the line is sent to chan, send a message to the next bot
02:33:20 <+madvirii> cuz it would get old editing your fav asciis
02:33:36 <@Jmax> system?
02:33:44 <+madvirii> yah like, read the txt file in line by line in an array, and then assign a line number to each bot accordingly, if they got banned, it might affect it, unless u design a failsafe, but it seems to be the right direction to head in
02:34:50 <@Jmax> if a bot can't send the message, it's delegated

2.1.2 Jmax, vx‘, and madvirii.

02:44:38 <@Jmax> if you mean *cat*—(1)ting, then here's what I have in mind:
02:44:41 <@Jmax> 1) determine latency
02:44:47 < vx‘> if it's gonna have the ability to scroll an ascii with the whole set of bots or some that ie. aren't banned on the channel
02:44:50 <@Jmax> 2) ignore any bots with high latency, if there's enough bots
02:44:56 < vx‘> then you wouldn't want some bots ruining the ascii because of slow links
02:45:38 <+madvirii> well, if we determine latency, and then just have the ten fastest bots
02:45:52 <@Jmax> no, we can't just ignore the bots
02:45:59 < vx‘> hardly efficient
02:46:03 <@Jmax> if we only have 10, then it'll still be limited. esp. if we have

100 more
02:46:34 < vx‘> apart from that, low latency links might get a) hit by the other bots scrolling b) throttled c) {banned,muted,shunned,glined}
02:47:29 <@Jmax> 3) determine which bots are capable of speaking in the channel. (not{muted,banned,shunned}, etc.)
02:47:56 < vx‘> freenode’s ircd might have some gay features to suppress ruining, and you don’t necessarily get numerics from the ircd informing you
02:48:11 <+madvirii> that is a good 3-step process to execute *prior* to even attempting to load a file
02:48:31 <@Jmax> 4) pull the file into an array, and assign each line to one of those bots
02:48:51 <@Jmax> 5) measure nick length, pad accordingly
02:50:18 < vx‘> but there are some time constraints on that code, proportionally to the amount of bots you’re scrolling with
02:50:28 <@Jmax> actually, in step 4, do not assign each line to a bot, maintain each list separately, a queue for {bot,line}s
02:55:02 <@Jmax> 6) issue the first line to the first bot in queue, and make the second bot wait for a successful message. If, during that time, it is determined by the first bot that it *cannot* send the message (and it’s not a latency issue), the line is re-assigned to the next bot in the queue and removed from the queue.
02:55:34 <@Jmax> 7) repeat until the file is complete.
02:55:52 <@Jmax> that will ensure that, if a bot is kicked, the rest of the bots take account for it
02:56:04 <+madvirii> so the number of lines in the file queue will determine the number of bot instances in the bot queue
02:56:05 <@Jmax> and any lines assigned to that bot are not ignored
02:56:10 <@Jmax> no, completely independent
02:56:27 < vx‘> what do you in case of a line not arriving due to ie. unexpected network problems, the link going down completely, etc.
02:56:39 <+madvirii> ok so its just for lines in queue
02:57:24 <@Jmax> well, that’s what the second part of step 6 is for, but that doesn’t account for netsplits
02:57:48 < vx‘> or any other network issues for that matter, like the proxy going down or lagging, ...
03:00:02 <@Jmax> sure it does, there will be a timeout, there will be the chance that the bot is just very very lagged, and wasn’t removed from the queue earlier, and sends the message, reaches the timeout, and the other bot replaces it, however, the message was still sent, so it will show up later
03:02:08 < vx‘> sure, but out of order and if that’d happen too often it’d ruin the whole thing
03:02:16 <@Jmax> right: i think that will be a rarity

2.2 LiteralKa blogging to no-one in particular.

13:51:56 <&LiteralKa> Maybe a command to spam a specific type of person in a given channel: (non-){ops,ircops,voiced}, all, etc.
13:53:34 <&LiteralKa> Synchronized ASCII flooding would be p. cool (read: *cat*(1)-style file flooding.)
13:53:48 <&LiteralKa> Like, delegate `x` amount of lines to each bot based on connection speed or some other algorithm or something.

2.3 Rufas

2.3.1 Rufas, sparc, and thyme

21:38:16 <+Rucas> i highly suggest you check out `STUPID`, run it from a very fast box and it can DDOS an ircd just by flooding text in chat, `SSH Tunnel Utilizing Python IRC Destroyer`
21:38:54 < sparc> *Rucas*: how fast of a box are we talking
21:39:00 <+Rucas> 100mbit is perfect, basically you need enough bandwidth to push all the OTHER boxes at a decent rate, it still does pretty well from a standard cable line though
21:39:42 < thyme> cant do something like: start remote processes on machines to avoid having to have one big monsterbox and just send signalling from home box
21:40:09 <+Rucas> well the original intent was cooperative flooding so like, you'd have 5 hosts and they'd all paste lines of one ascii, so you could spam asciis but bots wouldn't molish you but i never got around to that, because of *jenk*'s `irc-rc`

2.3.2 Rufas, incog, and rshxd.

21:42:43 < rshxd> *Rucas*: since when did you write `ASIAN`
21:42:50 <&Rucas> fucing years ago
21:42:59 < rshxd> I thought *abez* wrote that for you
21:43:02 <&Rucas> no, i wrote it, and *abez* rewrote part of it
21:43:04 < incog> before that was `AYSYN`
21:43:13 <&Rucas> and then *Jmax* rewrote that, and it is on `ASIANv6` or some shit, but i wrote `ASIANv1`
21:43:36 < incog> was it based off *mef*'s shit or all new?
21:43:42 <&Rucas> all new, i wish i had *mef*'s code, but now i wrote `STUPID` which is much nicer than `ASIAN`

2.4 The l0de Radio Hour

2.4.1 Rufas at [S1E12] 16:26

l0de: [If you steal that idea] my lawyers will fuck you so severely that it's not even funny...
Rufas: I'm sure they will, just like that guy's 800 bot botnet running wild on

EFNet right now.

l0de: Holy shit, that's hillarious–he's got 800 bots now?

Rufas: Yeah, he's up to 900 now.

l0de: 900 fuckin' bots, through a JPEG redirect exploit-type of thing? [...] And for our listeners, basically this guy coded an Internet Explorer link that runs an mIRC exploit and basically **turns your computar into botnet**

Rufas: And he authorized us to reverse engineer it [...] We can use it to assrape *#politics*.

2.4.2 Rufas at [20100409] 30:00

l0de: Alright, so what people really want to hear about from you is the ruin: the mega ruin that is going on. And you have this **massive**- fucking botnet and you're annihilating everything that comes into contact with you. I know Hardchats was DDoSed massively last night, primarily because of your exploits. Why don't you tell us what happened?

Rufas: Well, not a heck of a lot has really changed with what I've been doing. I've used the new tool I wrote a few months ago called `STUPID` but-

l0de: Is this a reference to *Max Goldberg*'s `AYSYN`–no it was *mef* that wrote `AYSYN` right?

Rufas: *mef* wrote `AYSYN` and I wrote `ASIAN` based on that which was `Automated Synchronous IRC Attack Network`. And then based on that I made `STUPID`, which is `SSH Tunnel Utilizing Python IRC Destroyer`–because they all have to have some sort of stupid `GNU` acronym to go with them.

2.4.3 sloth at [20100409] 3:07:21

sloth: I've gotten a lot of comments about "Oh my God! This is the first IPv6 botnet I've ever seen."

sloth: We are on the cutting edge of IRC flooding: we are pushing it into IPv6 and people just **don't even know**.