



Open Source Threat Intelligence Chat Bot

Tony Lee | Sr. Technical Director
(Chief Minion Herder)

TONY@CYLANCE: ~ \$ WHOAMI



▪ 13 Years of Professional Security Experience

- Currently Sr. Technical Director at Cylance, Inc.

▪ Specialties

- Offensive security / Red teaming
- Rapid prototyping and product integration

▪ Education

- Bachelor's in Computer Engineering, Virginia Tech
- Master's in Security Informatics, Johns Hopkins University

▪ Research and Publications

- Contributing author to Hacking Exposed 7 and frequent blogger
- Wireless security, China Chopper web shell, Cisco's SYNful Knock router implant
- Forensic Investigator Splunk app



AGENDA

- Introduction
- Components
 - Software
 - Hardware
- Installation & Example Configuration
- Usage & Features
- Development
- Acknowledgements
- Demo Setup

INTRODUCTION

- Are you a SOC analyst or an incident responder?
- Do you get tired of pivoting to multiple tools to perform your investigation?
- Have you ever wanted a minion to do your research and bidding, but have limited time and budget?

Let **CyBot** be your minion!



COMPONENTS

▪ Software

- Linux, Mac, Windows
- Python3
- ErrBot
- CyBot Plugins from GitHub

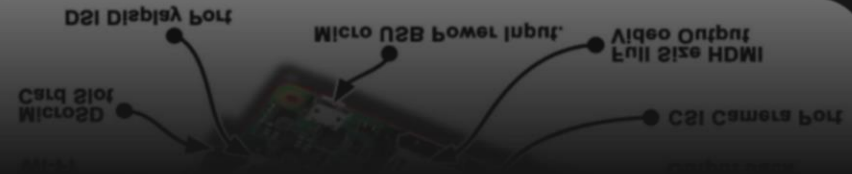
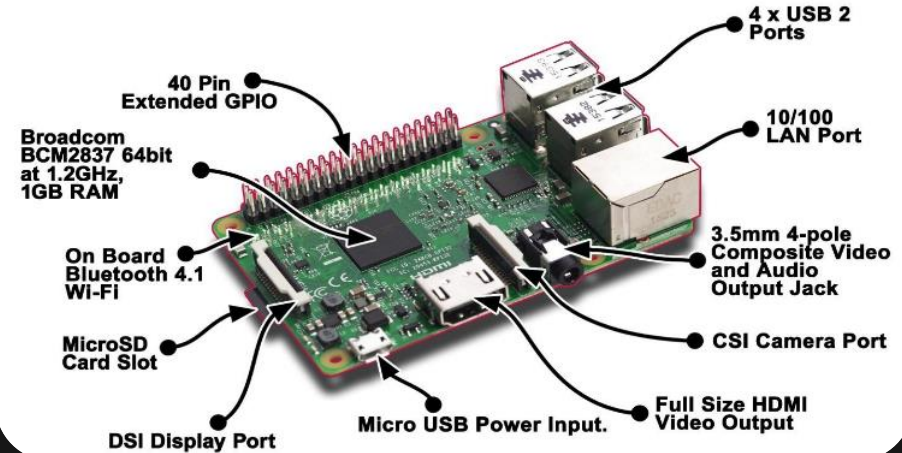


▪ Hardware

- VM or Unused hardware that is laying around
- or-
- Raspberry Pi
- Example:

https://www.amazon.com/gp/product/B01CUMNIV8/ref=oh_aui_detailpage_o01_s00?ie=UTF8&pssc=1

Raspberry PI 3



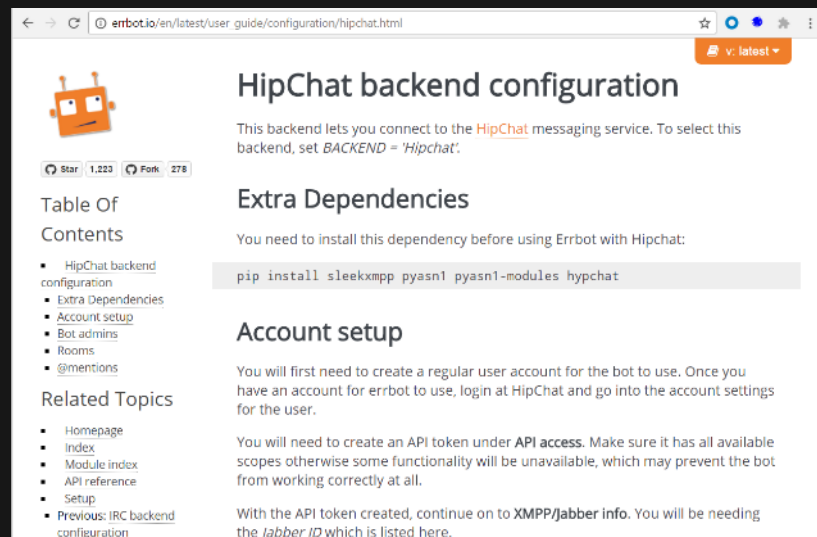
COMPONENTS

- **ErrBot is very flexible and supports multiple server backends**

- XMPP (Any standards-compliant XMPP/Jabber server should work - Google Talk/Hangouts included)
- Hipchat
- IRC
- Slack
- Telegram
- Tox (maintained separately)
- Gitter (maintained separately)
- CampFire (maintained separately)
- Skype (maintained separately)

- **Configuration largely depends on the chat protocol**

- Our example configuration will use HipChat
- Source: http://errbot.io/en/latest/user_guide/configuration/hipchat.html



The screenshot shows a web browser window with the URL `errbot.io/en/latest/user_guide/configuration/hipchat.html`. The page features the ErrBot logo (an orange robot head) and GitHub statistics (1,223 stars, 278 forks). The main heading is "HipChat backend configuration". Below it, a paragraph explains that this backend connects to the HipChat messaging service and that the `BACKEND` must be set to 'Hipchat'. A section titled "Extra Dependencies" states that a dependency must be installed before using ErrBot with HipChat, followed by a code block: `pip install sleekxmpp pyasn1 pyasn1-modules hypchat`. The "Account setup" section describes the need for a regular user account and an API token. A "Table Of Contents" on the left lists links to configuration, dependencies, account setup, bot admins, rooms, mentions, and related topics like the homepage, index, module index, API reference, and setup. A link to the "Previous: IRC backend configuration" is also present.

HipChat backend configuration

This backend lets you connect to the [HipChat](#) messaging service. To select this backend, set `BACKEND = 'Hipchat'`.

Extra Dependencies

You need to install this dependency before using Errbot with Hipchat:

```
pip install sleekxmpp pyasn1 pyasn1-modules hypchat
```

Account setup

You will first need to create a regular user account for the bot to use. Once you have an account for errbot to use, login at HipChat and go into the account settings for the user.

You will need to create an API token under **API access**. Make sure it has all available scopes otherwise some functionality will be unavailable, which may prevent the bot from working correctly at all.

With the API token created, continue on to [XMPP/Jabber info](#). You will be needing the *Jabber ID* which is listed [here](#).

Table Of Contents

- [HipChat backend configuration](#)
- [Extra Dependencies](#)
- [Account setup](#)
- [Bot admins](#)
- [Rooms](#)
- [@mentions](#)

Related Topics

- [Homepage](#)
- [Index](#)
- [Module index](#)
- [API reference](#)
- [Setup](#)
- [Previous: IRC backend configuration](#)

INSTALLATION

▪ Prerequisites

```
sudo apt-get install python3 python-dev libssl-dev python3-pip
```

```
sudo pip3 install errbot
```

```
mkdir ~/errbot-root
```

```
cd ~/errbot-root
```

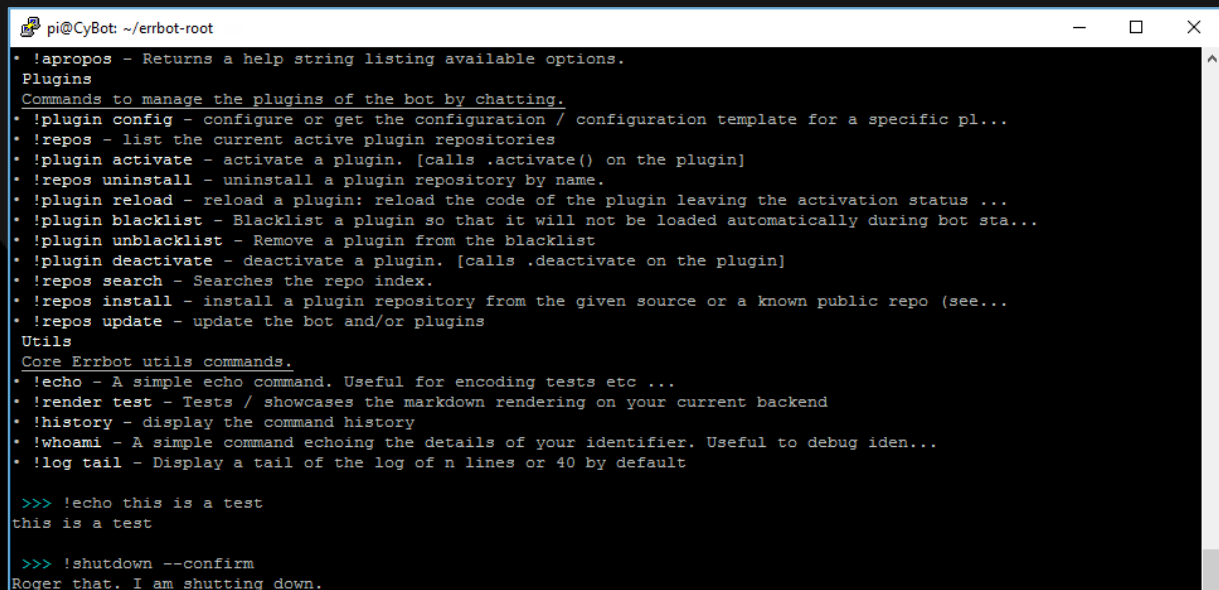
```
errbot --init
```

▪ Validation

```
errbot
```

```
!help
```

```
!shutdown --confirm
```



```
pi@CyBot: ~/errbot-root
* !apropos - Returns a help string listing available options.
Plugins
Commands to manage the plugins of the bot by chatting.
* !plugin config - configure or get the configuration / configuration template for a specific pl...
* !repos - list the current active plugin repositories
* !plugin activate - activate a plugin. [calls .activate() on the plugin]
* !repos uninstall - uninstall a plugin repository by name.
* !plugin reload - reload a plugin: reload the code of the plugin leaving the activation status ...
* !plugin blacklist - Blacklist a plugin so that it will not be loaded automatically during bot sta...
* !plugin unblacklist - Remove a plugin from the blacklist
* !plugin deactivate - deactivate a plugin. [calls .deactivate on the plugin]
* !repos search - Searches the repo index.
* !repos install - install a plugin repository from the given source or a known public repo (see...
* !repos update - update the bot and/or plugins
Utils
Core Errbot utils commands.
* !echo - A simple echo command. Useful for encoding tests etc ...
* !render test - Tests / showcases the markdown rendering on your current backend
* !history - display the command history
* !whoami - A simple command echoing the details of your identifier. Useful to debug iden...
* !log tail - Display a tail of the log of n lines or 40 by default

>>> !echo this is a test
this is a test

>>> !shutdown --confirm
Roger that. I am shutting down.
```

EXAMPLE CONFIGURATION

- Install ErrBot HipChat components

```
sudo pip3 install sleekxmpp pyasn1 pyasn1-modules hypchat
```

- Setup API user in HipChat Web UI

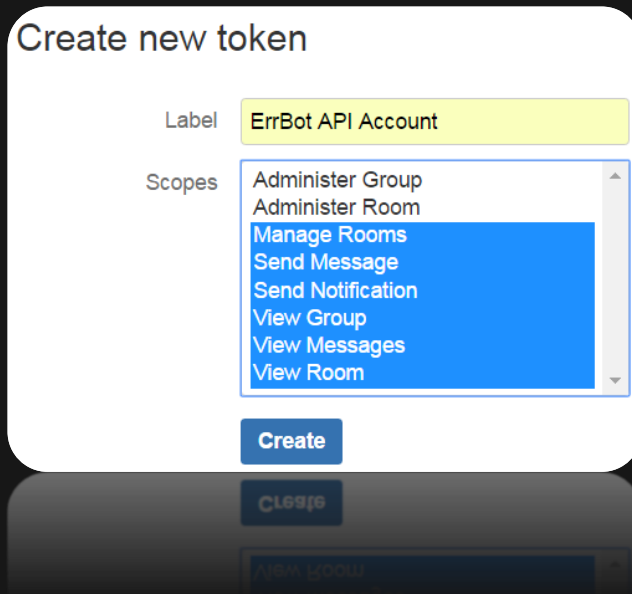
- Account Settings → API Access

- Create New Token

- Label: <API Label>

- Scopes:

- Select: Manage Rooms, Send Message,
Send Notification, View Group, View Messages, View Room



The screenshot shows a 'Create new token' dialog box. It has a 'Label' field with the value 'ErrBot API Account' and a 'Scopes' dropdown menu. The dropdown menu is open, showing a list of scopes: 'Administer Group', 'Administer Room', 'Manage Rooms', 'Send Message', 'Send Notification', 'View Group', 'View Messages', and 'View Room'. The 'Manage Rooms' scope is currently selected. At the bottom of the dialog, there are 'Create' and 'Close' buttons.

Create new token

Label: ErrBot API Account

Scopes: Administer Group, Administer Room, Manage Rooms, Send Message, Send Notification, View Group, View Messages, View Room

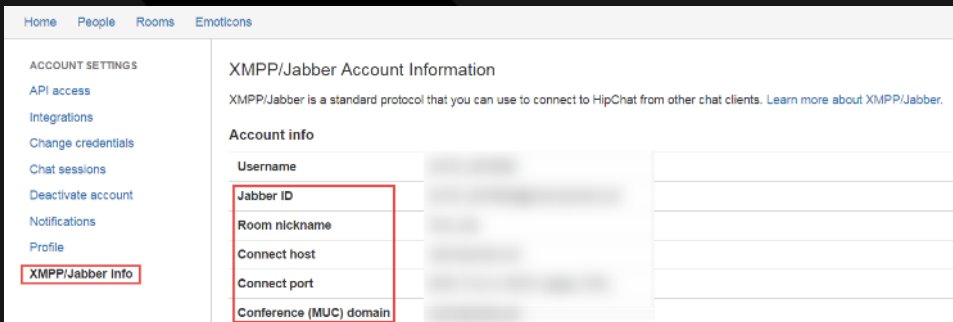
Create

Close

EXAMPLE CONFIGURATION

- Gather the following information from HipChat account settings

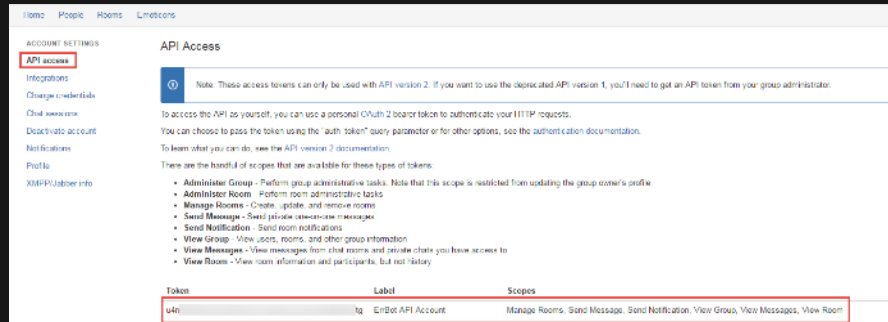
XMPP/Jabber Account Information



The screenshot shows the 'ACCOUNT SETTINGS' page with a sidebar on the left containing links: ACCOUNT SETTINGS, API access, Integrations, Change credentials, Chat sessions, Deactivate account, Notifications, Profile, and XMPP/Jabber Info (highlighted with a red box). The main content area is titled 'XMPP/Jabber Account Information' and includes a description: 'XMPP/Jabber is a standard protocol that you can use to connect to HipChat from other chat clients. Learn more about XMPP/Jabber.' Below this is the 'Account info' section with a table containing the following fields: Username, Jabber ID (highlighted with a red box), Room nickname, Connect host, Connect port, and Conference (MUC) domain (highlighted with a red box).

Account info	
Username	
Jabber ID	
Room nickname	
Connect host	
Connect port	
Conference (MUC) domain	

API token that we created



The screenshot shows the 'API Access' page. It includes a note: 'These access tokens can only be used with API version 2. If you want to use the deprecated API version 1, you'll need to get an API token from your group administrator.' Below this, it explains how to access the API and provides a list of scopes. A table at the bottom shows the details of the created API token, with the token value and its scopes highlighted by red boxes.

Token	Label	Scopes
4a1...	EnBot API Account	Manage Rooms, Send Message, Send Notification, View Group, View Messages, View Room

EXAMPLE CONFIGURATION

▪ Necessities for config.py

```
BACKEND = 'Hipchat' # defaults to XMPP
BOT_DATA_DIR = r'/root/errbot-root/data'
BOT_EXTRA_PLUGIN_DIR = '/root/errbot-root/plugins'
BOT_LOG_FILE = r'/root/errbot-root/errbot.log'
BOT_LOG_LEVEL = logging.INFO

BOT_IDENTITY = {
    'username' : '12345_123456@chat.hipchat.com',
    'password' : 'changeme',
    # Group admins can create/view tokens on the settings page after logging
    # in on HipChat's website
    'token' : 'ed4b74d628example312ff04', sleekxmpp pyasn1 pyasn1-modules hypchat
}

CHATROOM_PRESENCE = ()
CHATROOM_FN = 'CyBot'
```

EXAMPLE CONFIGURATION

- Added security for config.py

```
BOT_ADMINS = ('xxxxxxx@chat.hipchat.com',)

ACCESS_CONTROLS = {'status': {'allowrooms': ('someroom@conference.localhost',)},
                    'about': {'denyusers': ('*@evilhost',), 'allowrooms':
('room1@conference.localhost', 'room2@conference.localhost')},
                    'uptime': {'allowusers': BOT_ADMINS},
                    'help': {'allowmuc': False},
                    'help': {'allowmuc': False},
                    'ChatRoom:*': {'allowusers': BOT_ADMINS},
                    }

DIVERTE_TO_PRIVATE = ('help', 'about', 'status')
```

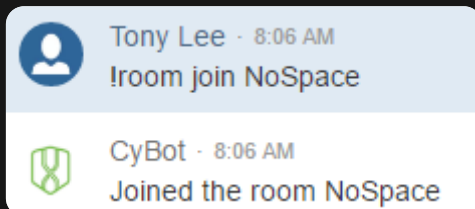
USAGE AND FEATURES

▪ Control CyBot (ErrBot specific commands)

Command	Arguments	Description
!help	N/A	Lists the help page
!room list	N/A	Lists the rooms that the bot is monitoring
!room join	XMPP room name	Joins a room to monitor
!room destroy	XMPP room name	Deletes a room – good for accidental creation
!restart	N/A	Useful for reloading plugins

▪ Notes:

- CyBot must be a member of the room to respond to commands
- Rooms with spaces in the name require the full HipChat name
 - !room join 12345_Room_With_Space_In_The_Name@conf.hipchat.com
 - vs.
 - !room join NoSpace



USAGE AND FEATURES

▪ Current CyBot Plugin Features

Command	Arguments	Description
!commands	N/A	Lists all possible commands
!vt	<hash URL>	VirusTotal Query
!hashid	<hash>	Identifies a hash type
!safebrowsing	<URL>	Google Safebrowsing Lookup
!whois	<domain or IP>	WHOIS Query
!nslookup	<domain or IP>	Forward and Reverse DNS Lookups
!geoip	<FQDN IP>	Perform GeoIP lookup of hosts
!unshorten	<Shortened URL>	
!linkextractor	<FQDN IP>	Extracts links from a site
!urldecode	<URL>	Decodes an encoded URL
!unixtime	<epoch>	Convert Unix time to human readable
!codename	N/A	Generates a 2 word project codename

USAGE AND FEATURES



Tony Lee · Jul-18 11:42 PM

!commands



CyBot · Jul-18 11:42 PM

== File commands ==

!vt <hash> - VirusTotal Query (ex: 57f222d8fbe0e290b4bf8eaa994ac641)

!hashid <hash> - Identify a hash type (e.g. MD5, SHA1) (Props: c0re)

== Network commands ==

!vt <URL> - VirusTotal Query

!safebrowsing <URL> - Google Safebrowsing lookup (ex: ihaveaproblem.info) (Props: Google)

!whois <domain> - WHOIS Query (ex: cylance.com) (Props: hackertarget.com)

!nslookup <FQDN|IP> - DNS forward/reverse Query (ex: www.cylance.com)

!geoip <FQDN|IP> - Perform GeoIP lookup of host (ex: www.cylance.com) (Props: freegeoip.net)

!unshorten <shortened URL> - Unshortens URLs (ex: goo.gl/IGL1IE)

!linkextractor <FQDN|IP> - Extracts links from a site and safely displays them (ex: <https://www.google.com>)

!urldecode <url> - Decodes encoded URLs (ex: [/%75%72%6C?%73%61=%74](http://%75%72%6C?%73%61=%74))

== Misc ==

!unixtime <epoch> - Convert Unix time to human readable (ex: 1347517370)

!codename - Generates a 2 word project codename (Props: Mark Biek)

[Show less](#)

2 word j622

DEVELOPMENT

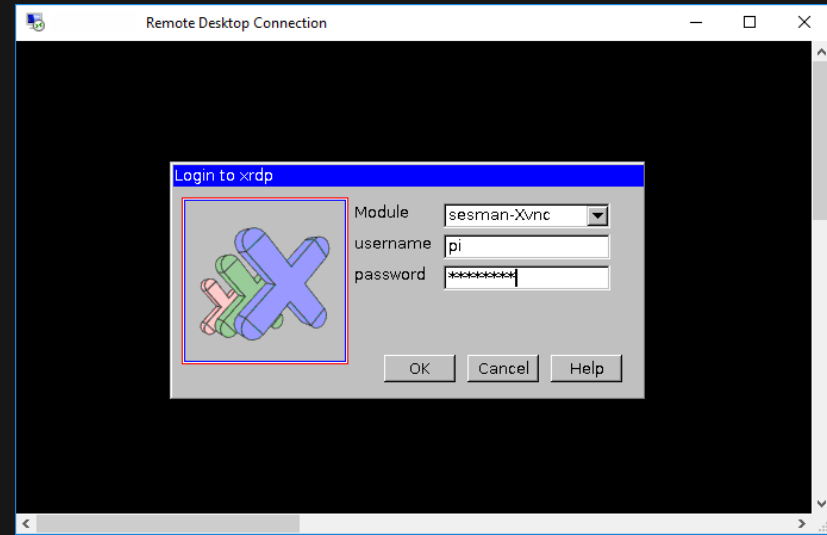
- Always on-going
- Roadmap is robust
 - Automated report writing
 - SIEM integration
- Open to everyone as a community project
- Plugins written in Python
- Plugins are chat backend agnostic and will work with all platforms supported by errbot
- Code contributions go to GitHub
- New ideas welcomed!



PRO TIPS

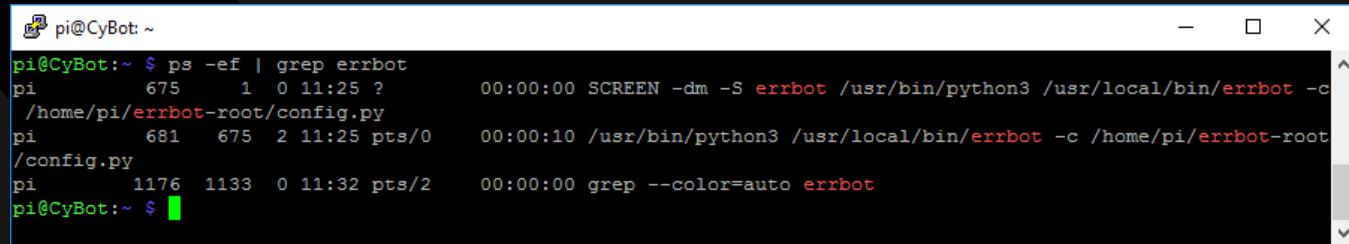
- If installed on Raspberry Pi
 - Set a static IP
 - `sudo leafpad /etc/dhcpd.conf` (at the end enter)

```
interface eth0
static ip_address=<ip.ad.dr.ess>
static routers=<ip.ad.dr.ess>
static domain_name_servers=<ip.ad.dr.ess>
```
 - You may need to set US (or other regional) keyboard since this defaults to UK
 - “Start” -> Preferences -> Mouse and Keyboard Settings -> Keyboard tab -> Keyboard Layout -> United States -> English (US)
 - Install remote desktop
 - `sudo apt-get install vnc4server`
 - `sudo apt-get install xrdp`
 - Set SSH to autorun on boot
 - `sudo update-rc.d ssh enable`



PRO TIPS

- If installed on Raspberry Pi (cont.)
 - Set errbot to autorun on boot in a detached screen session
 - `sudo apt-get install screen`
 - `sudo leafpad /etc/rc.local`
 - Add the following line before the “exit 0” line:
 - `su - pi -c “screen -dm -S errbot /usr/bin/python3 /usr/local/bin/errbot -c /home/pi/errbot-root/config.py”`
 - Interact with the screen session by using: “screen -d -r”
 - Adjust screensaver settings easily (this will place a shortcut in preferences):
 - `sudo apt-get install xscreensaver`

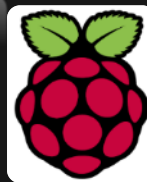
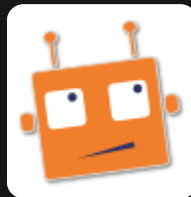


```
pi@CyBot: ~  
pi@CyBot:~ $ ps -ef | grep errbot  
pi      675      1  0 11:25 ?        00:00:00 SCREEN -dm -S errbot /usr/bin/python3 /usr/local/bin/errbot -c  
/home/pi/errbot-root/config.py  
pi      681      675  2 11:25 pts/0    00:00:10 /usr/bin/python3 /usr/local/bin/errbot -c /home/pi/errbot-root  
/config.py  
pi      1176    1133  0 11:32 pts/2    00:00:00 grep --color=auto errbot  
pi@CyBot:~ $
```

ACKNOWLEDGEMENTS

- **Platform:**

- ErrBot - <http://errbot.io/en/latest/>
- Raspberry Pi - <https://www.raspberrypi.org/>



- **Plugins:**

- VirusTotal - <https://www.virustotal.com/>
- Freegeoip – freegeoip.net
- Google Safebrowsing - <https://developers.google.com/safe-browsing/>
- Hashid – c0re
- Unshorten – unshorten.me

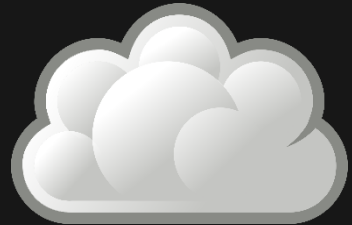
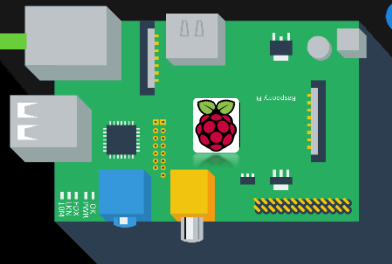


Google Safe Browsing

- **People:**

- Bill Hau, Corey White, Dennis Hanzlik, Ian Ahl, Dave Pany, Dan Dumond, Kyle Champlin

DEMO!



The background is a blurred photograph of a large crowd of people, possibly at a conference or event. Overlaid on this are several semi-transparent geometric shapes: a large dark grey triangle on the left, a lighter grey triangle at the top, and a dark red triangle at the bottom left. The text is centered in the middle of the image.

QUESTIONS — AND — ANSWERS



Thank you

BACKUP SLIDE IN CASE LIVE DEMO TANKS

```
>>> !geoip www.google.com
"ip": "2607:f8b0:4006:81b::2004"
"country_code": "US"
"country_name": "United States"
"region_code": ""
"region_name": ""
"city": ""
"zip_code": ""
"time_zone": ""
"latitude": 37.751
"longitude": -97.822
"metro_code": 0
```

```
>>> !hashid 57f222d8fbe0e290b4bf8eaa994ac641
Analyzing '57f222d8fbe0e290b4bf8eaa994ac641'
[+] MD2
[+] MD5
[+] MD4
[+] Double MD5
[+] LM
[+] RIPEMD-128
[+] Haval-128
[+] Tiger-128
[+] Skein-256(128)
--snip--
```

```
>>> !safebrowsing ihaveaproblem.info
{'matches': [{'threatEntryType': 'URL', 'cacheDuration': '300s', 'threat': {'url': 'ihaveaproblem.info'}, 'threatType': 'SOCIAL_ENGINEERING', 'platformType': 'ANY_PLATFORM']}]}
```