Author: Duong Tran My Linh

Published date: 18 May 2023

Build your own penetration testing lab with AWS, Kali Linux and OWASP ZAP

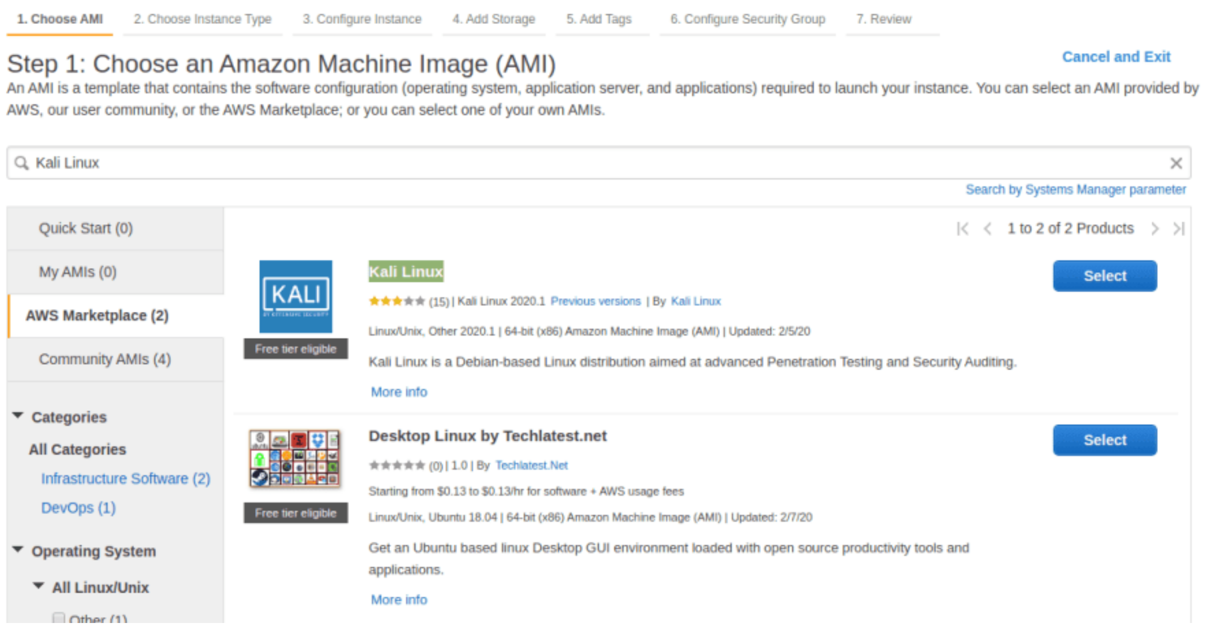
EC2 and Kali Linux

EC2 stands for Elastic Compute Cloud. In our case we'll use EC2 instance to get our Kali image installed, up and running.

Kali Linux is a Linux distribution based on Debian. It's commonly designed for penetration testing purposes, and has a lot of useful tools preinstalled.

Installation steps

- Login to your AWS account.
- On the home page, either search or select the EC2 service that is located under the Compute category.
- On the EC2 Dashboard page, click the "Launch instance" instance button.
- We're looking for Kali Linux images, so, on the left sidebar, choose "AWS Marketplace", then use the search input field at the top - enter "Kali Linux" and hit the Enter/Return key on your keyboard.
- You should see the "Kali Linux" image in the search results.



Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

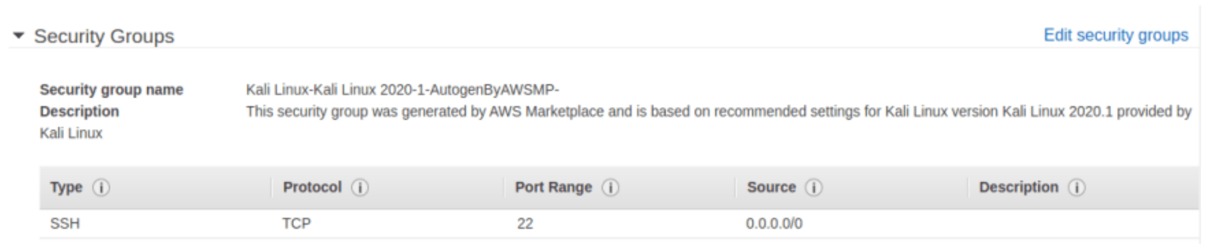
Kali Linux

Kali Linux
★★★★★ (15) | Kali Linux 2020.1 Previous versions | By Kali Linux
Linux/Unix, Other 2020.1 | 64-bit (x86) Amazon Machine Image (AMI) | Updated: 2/5/20
Kali Linux is a Debian-based Linux distribution aimed at advanced Penetration Testing and Security Auditing.

Desktop Linux by Techlatest.net
★★★★★ (0) | 1.0 | By Techlatest.Net
Starting from $0.13 to $0.13/hr for software + AWS usage fees
Linux/Unix, Ubuntu 18.04 | 64-bit (x86) Amazon Machine Image (AMI) | Updated: 2/7/20
Get an Ubuntu based linux Desktop GUI environment loaded with open source productivity tools and applications.

- Click the appropriate "Select" button.
- In the opened modal you see the description of Kali Linux and Pricing Details. Click "Continue".

- At this moment I don't care about best experience, so I choose "t2.micro" just because the free tier is eligible. Choose the instance type and click "Review and Launch". Clicking the "Review and Launch" button will redirect us to the last step. AWS has some default configuration setup for storage, tags, security groups etc. You can edit those configs if you want.
- On the "Step 7: Review Instance Launch" page you can review your instance configuration. Please make sure your security group allows SSH.



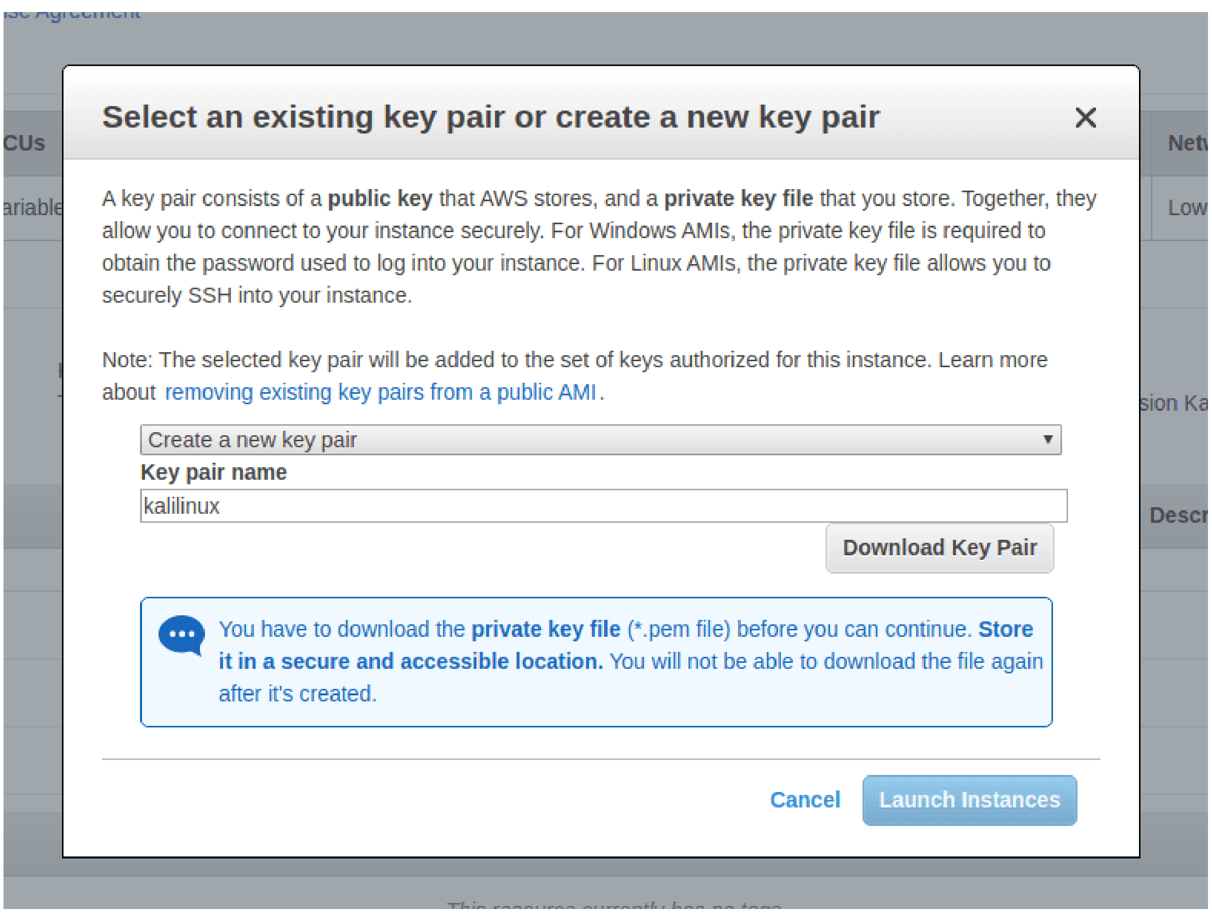
Security Groups

Edit security groups

Security group name Kali Linux-Kali Linux 2020-1-AutogenByAWSMP-
Description This security group was generated by AWS Marketplace and is based on recommended settings for Kali Linux version Kali Linux 2020.1 provided by Kali Linux

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	0.0.0.0/0	

- Click the "Launch" button.
- In the opened modal select the "Create a new key pair" option in the dropdown field, enter a key pair name (e.g. kalilinux), and click the "Download Key Pair" button.



Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI.

Create a new key pair

Key pair name

kalilinux

You have to download the **private key file** (*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

- Please make sure you have downloaded the key pair file, and click the "Launch instance" button.

That's all! Few moments later you should see a new running ec2 instance - just click the "Instances" link in the left menu.

Setup automatic scan with OWASP ZAP

Connection

I'm running Kali on AWS so I want to connect to the instance using SSH.
I have the `.pem` file, so I need to run just few commands.

```
sudo chmod 400 kali.pem
ssh -i kali.pem ec2-user@your-public-dns
```

Installation

I expected to have `zaproxy` preinstalled, but no. So, let's install it. Though I've installed the 2019.4 version of Kali.
Let's run the command and get the `zaproxy` installed:

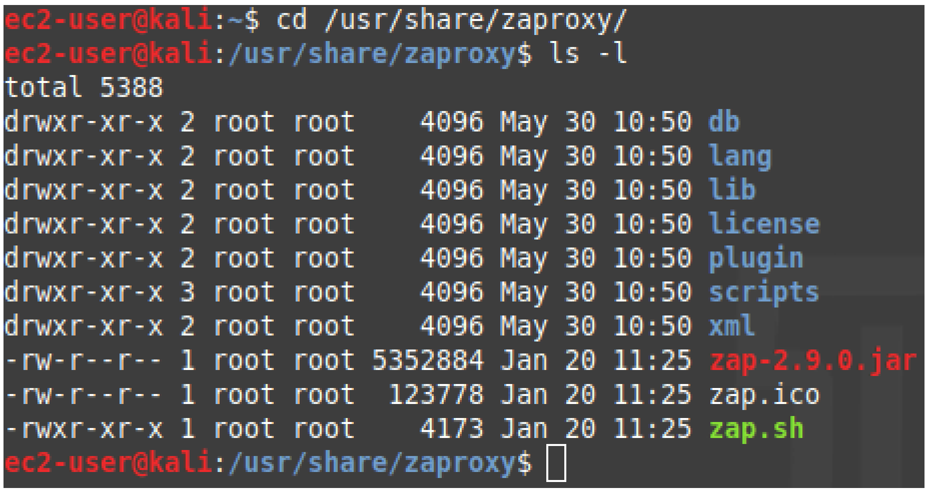
```
sudo apt-get update && sudo apt-get install zaproxy
```

Hopefully you've completed the installation successfully.
If you run the command `zaproxy`, you should probably see output like this:

```
Found Java version 11.0.5
Available memory: 982 MB
Using JVM args: -Xmx245m
0 [main] INFO org.zaproxy.zap.GuiBootstrap  - OWASP ZAP 2.9.0 started
30/05/2020, 14:57:21 with home /home/ec2-user/.ZAP/
2 [main] FATAL org.zaproxy.zap.GuiBootstrap  - ZAP GUI is not supported
on a headless environment.
Run ZAP inline or in daemon mode, use -help command line argument for
more details.
ZAP GUI is not supported on a headless environment.
Run ZAP inline or in daemon mode, use -help command line argument for
more details.
```

We're using zap on a headless environment, so let's figure out how to use this tool in command line.

For some reason `zaproxy -cmd -help` command didn't work for me, so I had to figure out another way to run the tool.
The `whereis zaproxy` command shows us the following output `zaproxy: /usr/bin/zaproxy /usr/share/zaproxy`.
We're looking for `zap.sh` file located at `/usr/share/zaproxy` directory.



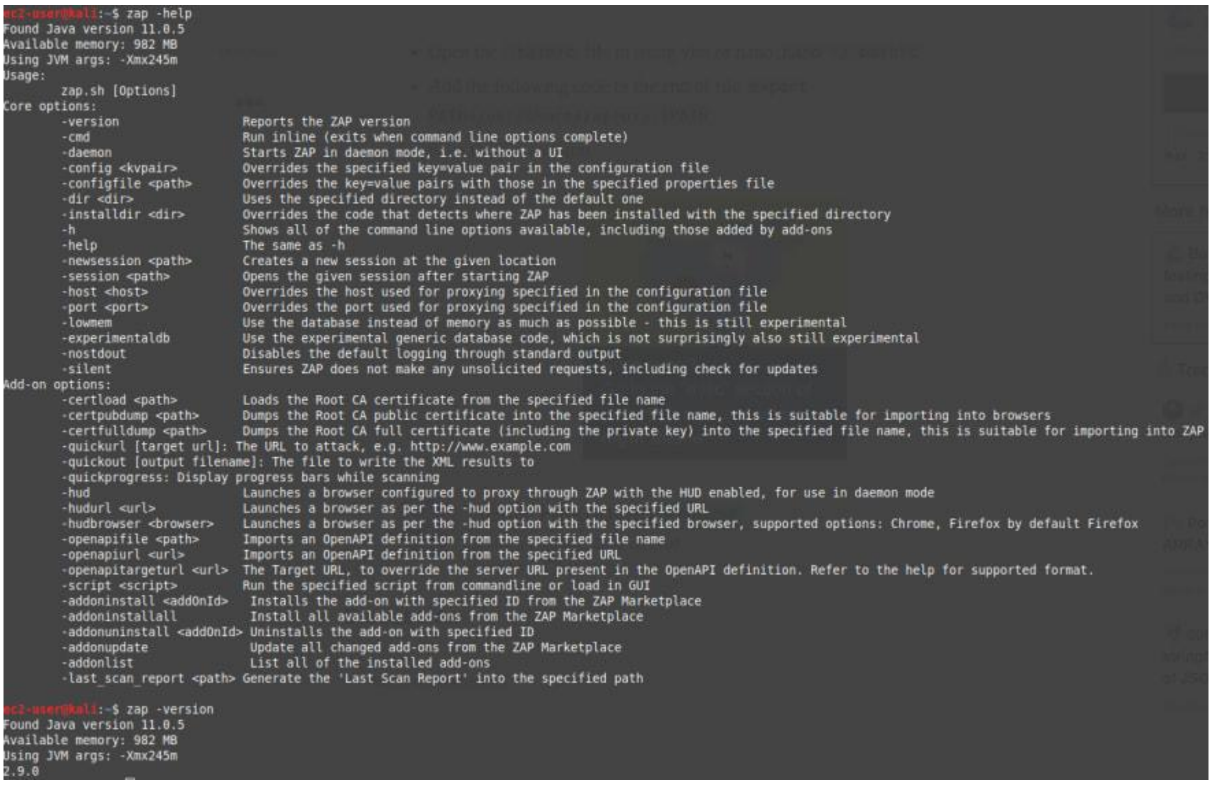
```
ec2-user@kali:~$ cd /usr/share/zaproxy/
ec2-user@kali:/usr/share/zaproxy$ ls -l
total 5388
drwxr-xr-x 2 root root    4096 May 30 10:50 db
drwxr-xr-x 2 root root    4096 May 30 10:50 lang
drwxr-xr-x 2 root root    4096 May 30 10:50 lib
drwxr-xr-x 2 root root    4096 May 30 10:50 license
drwxr-xr-x 2 root root    4096 May 30 10:50 plugin
drwxr-xr-x 3 root root    4096 May 30 10:50 scripts
drwxr-xr-x 2 root root    4096 May 30 10:50 xml
-rw-r--r-- 1 root root 5352884 Jan 20 11:25 zap-2.9.0.jar
-rw-r--r-- 1 root root  123778 Jan 20 11:25 zap.ico
-rwxr-xr-x 1 root root    4173 Jan 20 11:25 zap.sh
ec2-user@kali:/usr/share/zaproxy$ 
```

You can simply run it with `bash /usr/share/zaproxy/zap.sh` command.

Making a globally available command `zap`

If you're too lazy to type as many characters, then you can make an alias `zap` to `/usr/share/zaproxy/zap.sh`
To do that, we need to perform few simple steps and edit the `.bashrc` file.

- Open the `.bashrc` file using vim or nano - `nano ~/.bashrc`
- Add the following code to the end of file - `alias zap="bash /usr/share/zaproxy/zap.sh"`
- Save the file and quit
- Run `source ~/.bashrc` to apply changes, otherwise you need to log out and log in again
- Run `zap -help` or `zap -version`



```
ec2-user@kali:~$ zap -help
Found Java version 11.0.5
Available memory: 982 MB
Using JVM args: -Xmx245m
Usage:
        zap.sh [Options]
Core options:
        -version                Reports the ZAP version
        -cmd                    Run inline (exits when command line options complete)
        -daemon                 Starts ZAP in daemon mode, i.e. without a UI
        -config <kvpair>        Overrides the specified key=value pair in the configuration file
        -configfile <path>      Overrides the key=value pairs with those in the specified properties file
        -dir <dir>              Uses the specified directory instead of the default one
        -installdir <dir>       Overrides the code that detects where ZAP has been installed with the specified directory
        -h                      Shows all of the command line options available, including those added by add-ons
        -help                   The same as -h
        -newsession <path>      Creates a new session at the given location
        -session <path>         Opens the given session after starting ZAP
        -host <host>            Overrides the host used for proxying specified in the configuration file
        -port <port>            Overrides the port used for proxying specified in the configuration file
        -lowmem                 Use the database instead of memory as much as possible - this is still experimental
        -experimentaldb         Use the experimental generic database code, which is not surprisingly also still experimental
        -nostdout               Disables the default logging through standard output
        -silent                 Ensures ZAP does not make any unsolicited requests, including check for updates
Add-on options:
        -certload <path>        Loads the Root CA certificate from the specified file name
        -certpubdump <path>     Dumps the Root CA public certificate into the specified file name, this is suitable for importing into browsers
        -certfulldump <path>    Dumps the Root CA full certificate (including the private key) into the specified file name, this is suitable for importing into ZAP
        -quickurl [target url]: The URL to attack, e.g. http://www.example.com
        -quickout [output filename]: The file to write the XML results to
        -quickprogress: Display progress bars while scanning
        -hud                    Launches a browser configured to proxy through ZAP with the HUD enabled, for use in daemon mode
        -hudurl <url>           Launches a browser as per the -hud option with the specified URL
        -hudbrowser <browser>   Launches a browser as per the -hud option with the specified browser, supported options: Chrome, Firefox by default Firefox
        -openapifile <path>     Imports an OpenAPI definition from the specified file name
        -openapiurl <url>       Imports an OpenAPI definition from the specified URL
        -openapitargeturl <url> The Target URL, to override the server URL present in the OpenAPI definition. Refer to the help for supported format.
        -script <script>        Run the specified script from commandline or load in GUI
        -addoninstall <addOnId>  Installs the add-on with specified ID from the ZAP Marketplace
        -addoninstallall         Install all available add-ons from the ZAP Marketplace
        -addonuninstall <addOnId> Uninstalls the add-on with specified ID
        -addonupdate             Update all changed add-ons from the ZAP Marketplace
        -addonlist               List all of the installed add-ons
        -last_scan_report <path> Generate the 'Last Scan Report' into the specified path

ec2-user@kali:~$ zap -version
Found Java version 11.0.5
Available memory: 982 MB
Using JVM args: -Xmx245m
2.9.0
```

As you can see I'm using version 2.9.0.
If your output is similar to mine, then we're done here! 🚀

Scan

Now we are ready to execute our first scan. Simply, run the following command:

```
zap -cmd -quickurl http://example.com -quickprogress -quickout ~/out.xml
```

Replace the "example.com" with whatever host you want to scan.
Here is my console output:

```
ec2-user@kali:~$ zap -cmd -quickurl http://example.com -quickprogress
-quickout ~/out.xml
Found Java version 11.0.5
Available memory: 982 MB
Using JVM args: -Xmx245m
Accessing URL
Using traditional spider
Active scanning
[===================] 100%
Attack complete
Writing results to /home/ec2-user/out.xml
```

So, we just ran an attack on `example.com` host and got the output in XML format - the out.xml file located in `/home/ec2-user` directory.
Good start. But there is a one problem - I don't want output to be in XML format. I want PDF!

Add-ons

There are lot of useful add-ons in the ZAP Marketplace. We need the one named "Export Report".
ZAP allows us to install add-ons by their ID. Let's install the add-on:

```
zap -cmd -addoninstall exportreport
```

What's next?

In the next post I want to figure out the usage of Export Report add-on.
In the end I want to have scheduled scans running automatically and generating me nice PDF reports.

Intro

In the last post I described the web application scanning with Zaproxy CLI installed on Kali Linux.
Zaproxy by default generates an output in XML format. It's cool, but I want a PDF report.
I've installed the Export Report add-on that provides an ability to generate nice reports in different formats (`.xhtml`, `.pdf`, `.json`, `.xml`).

Scan & Export

Now, let's put together a simple bash script.

```
#!/bin/bash

# Getting the host passed as an argument to the script
host="$1"

# Getting current timestamp to use it in the session name
```

```
timestamp=$(date '+%s');

# Exit if host is not specified
if [ -z "$host" ]; then
    echo -e "Please pass the host argument.\r"
    exit 1
fi

# Launching the scan
/usr/share/zaproxy/zap.sh -quickurl "$host" -newsession "$timestamp"
-cmd;

# Defining variables that contain metadata for the report
report_name="Vulnerability Report - $host"
prepared_by="h4x0r"
prepared_for="X Corp"
scan_date=$(date -d @$timestamp)
report_date=$(date -d @$timestamp)
scan_version="N/A"
report_version="N/A"
report_description="Home page vulnerability report of the Example
project."
file_name="$timestamp"

# Getting the report generated in XHTML format
/usr/share/zaproxy/zap.sh -export_report "$HOME"/"$file_name".xhtml
-source_info
"$report_title;$prepared_by;$prepared_for;$scan_date;$report_date;$scan_
version;$report_version;$report_description" -alert_severity "t;t;f;t"
-alert_details "t;t;t;t;t;t;f;f;f;f" -session "$timestamp.session" -cmd
```

Here you can read about "Export Report" command line options.

PDF is not supported

Unfortunately the 2.9.0 version I've installed, does not support output in `.pdf` format. There was an issue and PR for that, so future releases probably will include that fix.

At this point I don't want to upgrade anything, just because I want to have "Latest release" instead of "Pre-release", so I have a workaround - WkHTMLtoPDF.

The Workaround

If you're from the future - you're probably able to generate PDF report without this workaround.
The idea is to get the report in XHTML format and convert it to PDF programmatically.

Installation

Because I'm using Kali, which is based on Debian, I need to download the `deb` package (check the Downloads section if you need something else).
`wget https://github.com/wkhtmltopdf/packaging/releases/download/0.12.6-rc/wkhtmltox_0.12.6-0.20200605.30.rc.faa06fa.stretch_amd64.deb`
Now install the package.

```
sudo dpkg -i wkhtmltox_0.12.6-0.20200605.30.rc.faa06fa.stretch_amd64.deb
```

Usage

To convert `html/xhtml` to `pdf`, simply run a command:

```
wkhtmltopdf file.xhtml file.pdf
```

Updating the bash script

Now let's convert the XHTML report to PDF using the `wkhtmltopdf` tool. We just need to include single line to the end of the script.

```
#!/bin/bash

# Get the host passed as an argument to the script
host="$1"

# Getting current timestamp to use it in the session name
timestamp=$(date '+%s');

# Exit if host is not specified
if [ -z "$host" ]; then
    echo -e "Please pass the host argument.\r"
    exit 1
fi

# Launching the scan
/usr/share/zaproxy/zap.sh -quickurl "$host" -newsession "$timestamp"
```

```
-cmd;

# Defining variables that contain metadata for the report
report_name="Vulnerability Report - $host"
prepared_by="h4x0r"
prepared_for="X Corp"
scan_date=$(date -d @$timestamp)
report_date=$(date -d @$timestamp)
scan_version="N/A"
report_version="N/A"
report_description="Home page vulnerability report of the Example
project."
file_name="$timestamp"

# Getting the report generated in XHTML format
/usr/share/zaproxy/zap.sh -export_report "$HOME"/"$file_name".xhtml
-source_info
"$report_title;$prepared_by;$prepared_for;$scan_date;$report_date;$scan_
version;$report_version;$report_description" -alert_severity "t;t;f;t"
-alert_details "t;t;t;t;t;t;f;f;f;f" -session "$timestamp.session" -cmd

# Converting XHTML report to PDF
wkhtmltopdf "$HOME"/"$file_name".xhtml "$HOME"/"$file_name".pdf
```

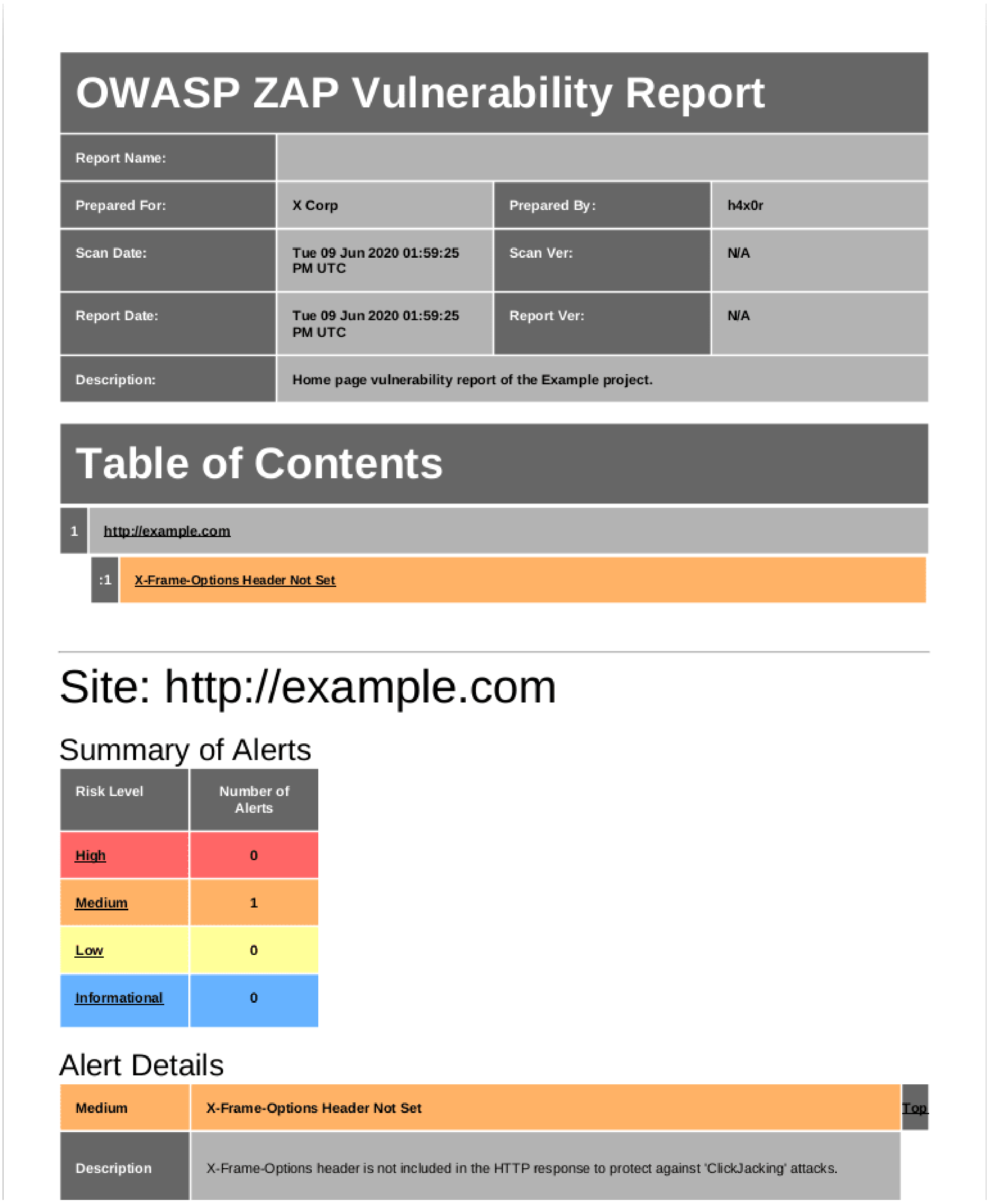
Hopefully my comments in the script are clear enough.
[Here is the Gist]

Though you can download it using `wget`.

```
wget
https://gist.githubusercontent.com/c0d3b0t/5139c61104f232206c4989770
1811b0d/raw/3de6356df81ed0a34310a095463e4c03ce2ee62a/run-zap.sh
```

You can launch that script by running `bash run-zap.sh http://example.com` where "example.com" is your target.

The generated report looks like this.



OWASP ZAP Vulnerability Report

Report Name:			
Prepared For:	X Corp	Prepared By:	h4x0r
Scan Date:	Tue 09 Jun 2020 01:59:25 PM UTC	Scan Ver:	N/A
Report Date:	Tue 09 Jun 2020 01:59:25 PM UTC	Report Ver:	N/A
Description:	Home page vulnerability report of the Example project.		

Table of Contents

1	http://example.com
:1	X-Frame-Options Header Not Set

Site: http://example.com

Summary of Alerts

Risk Level	Number of Alerts
High	0
Medium	1
Low	0
Informational	0

Alert Details

Medium	X-Frame-Options Header Not Set	Top
Description	X-Frame-Options header is not included in the HTTP response to protect against 'ClickJacking' attacks.	

So, now I've got same report in 2 different formats - XHTML and PDF. I've got XHTML exported by zap, and PDF converted using wkhtmltopdf.

What's next?

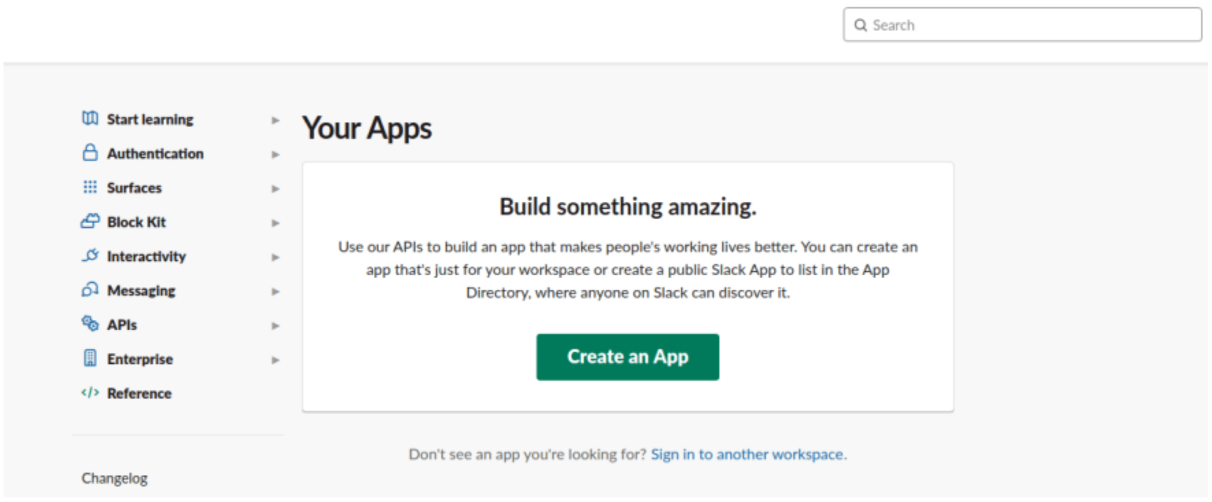
I need to think more about the automation process I want to implement. The process I have currently is:

1) run the bash script
2) download the report using `scp` or FileZilla
In my previous post I talked about exporting and generating PDF report from Zaproxy output.
Now I want to have my bash script updated to share that report to my Slack channel.

Publish a file on Slack

The Slack app

- Go to api.slack.com.
- Click the "Your Apps" link at the top right.
- Authenticate, if you're not authenticated.
- Click the "Create an App" button.



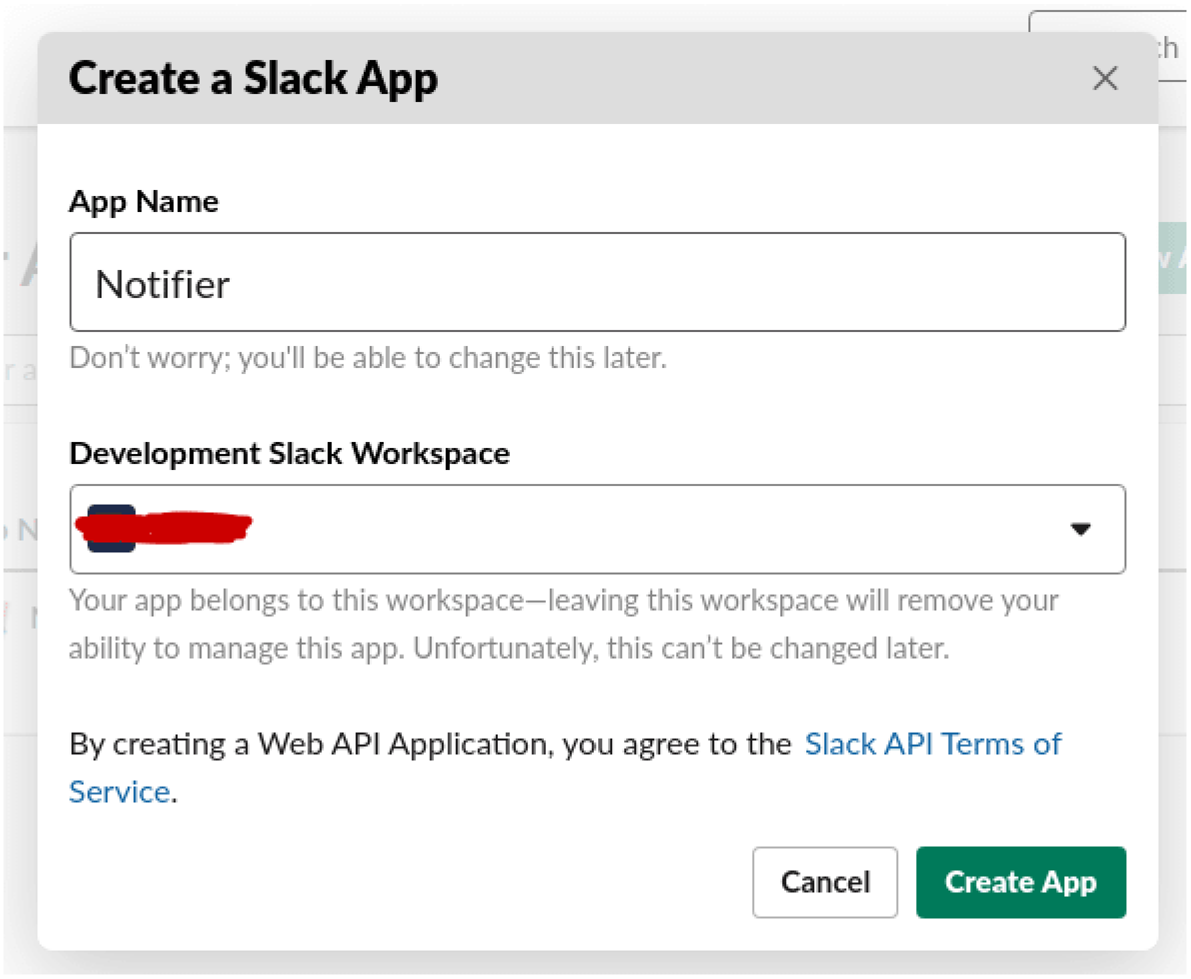
Your Apps

Build something amazing.

Use our APIs to build an app that makes people's working lives better. You can create an app that's just for your workspace or create a public Slack App to list in the App Directory, where anyone on Slack can discover it.

Don't see an app you're looking for? Sign in to another workspace.

- Set the App name, e.g. "Notifier", set the "Development Slack Workspace" and click the "Create App" button.



Create a Slack App

App Name

Notifier

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

Development Slack Workspace

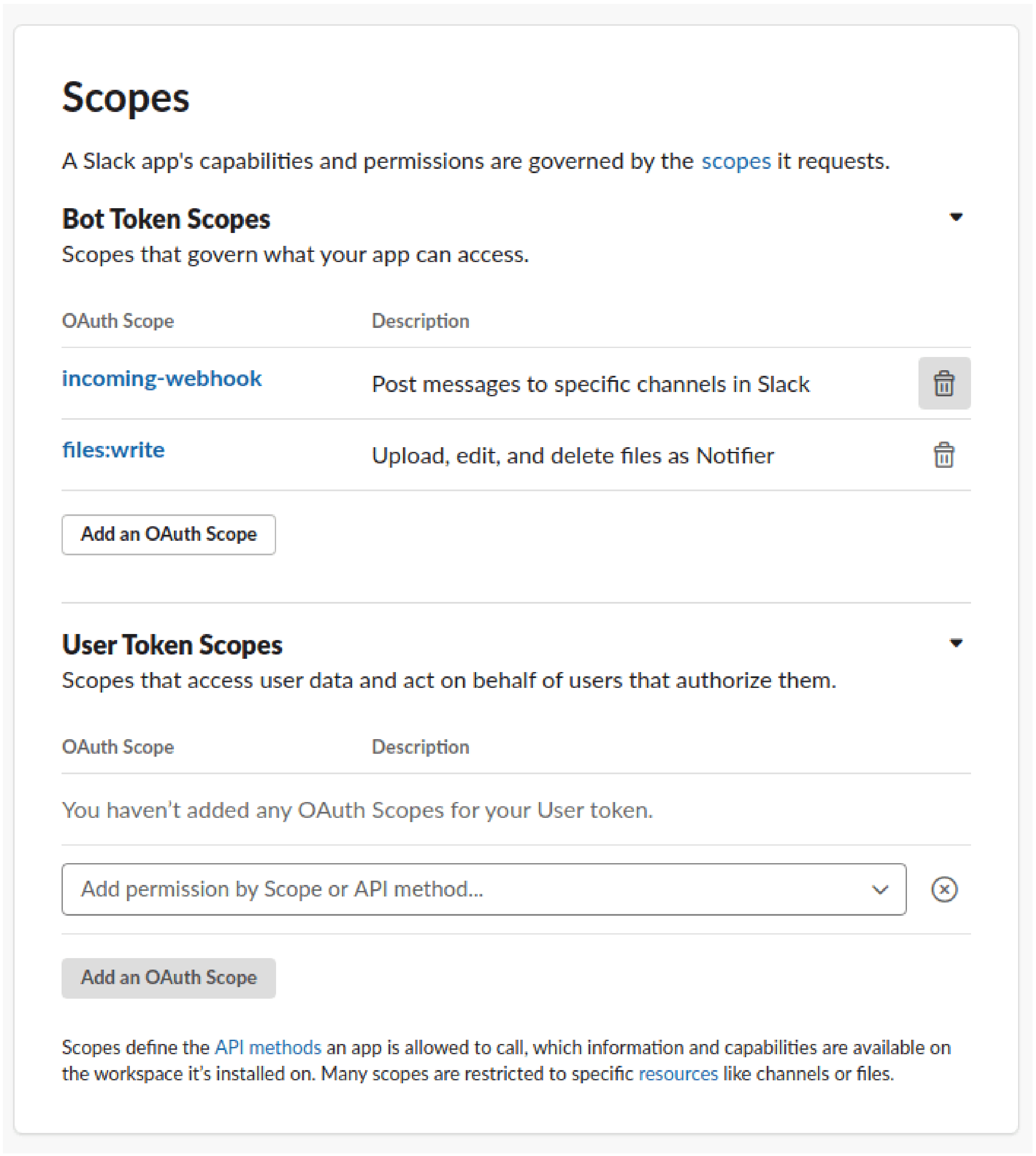
[redacted]

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.

Scopes and Permissions

- Click the "OAuth & Permissions" menu item in the "Features" section, or the "Permissions" item in the "Add features and functionality" section. Scroll down to the "Scopes" section and under the "Bot Token Scopes", click the "Add an OAuth Scope" button. In the opened dropdown search and select the "files:write" permission.



Scopes

A Slack app's capabilities and permissions are governed by the scopes it requests.

Bot Token Scopes

Scopes that govern what your app can access.

OAuth Scope	Description
incoming-webhook	Post messages to specific channels in Slack
files:write	Upload, edit, and delete files as Notifier

User Token Scopes

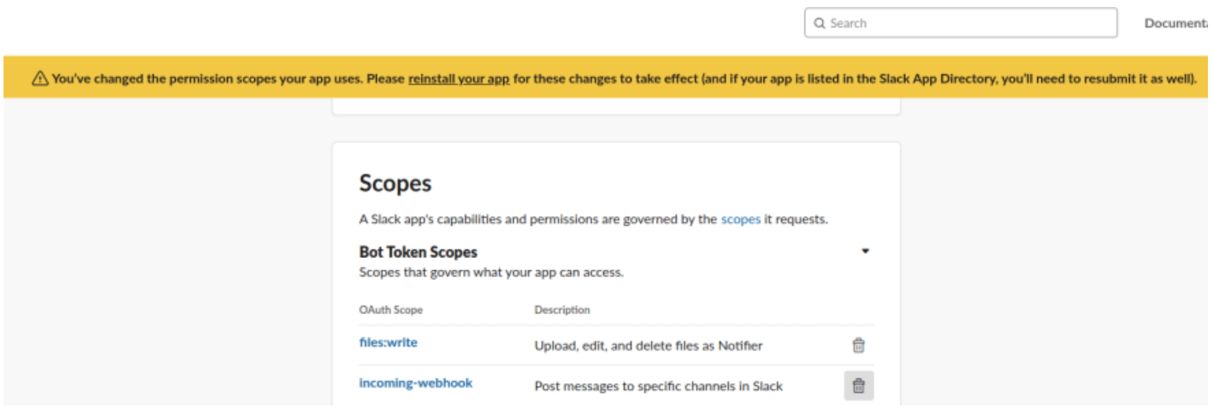
Scopes that access user data and act on behalf of users that authorize them.

OAuth Scope	Description

You haven't added any OAuth Scopes for your User token.

Scopes define the API methods an app is allowed to call, which information and capabilities are available on the workspace it's installed on. Many scopes are restricted to specific resources like channels or files.

In case if you're adding a scope to an existing Slack app that is installed to workspace (e.g. you may already have a Slack app for incoming webhooks), a big yellow alert may ask you to reinstall the app to apply changes. Click the "reinstall your app" link, specify a Slack channel, and click the "Allow" button.


Search

Document

⚠ You've changed the permission scopes your app uses. Please reinstall your app for these changes to take effect (and if your app is listed in the Slack App Directory, you'll need to resubmit it as well).

Scopes

A Slack app's capabilities and permissions are governed by the scopes it requests.

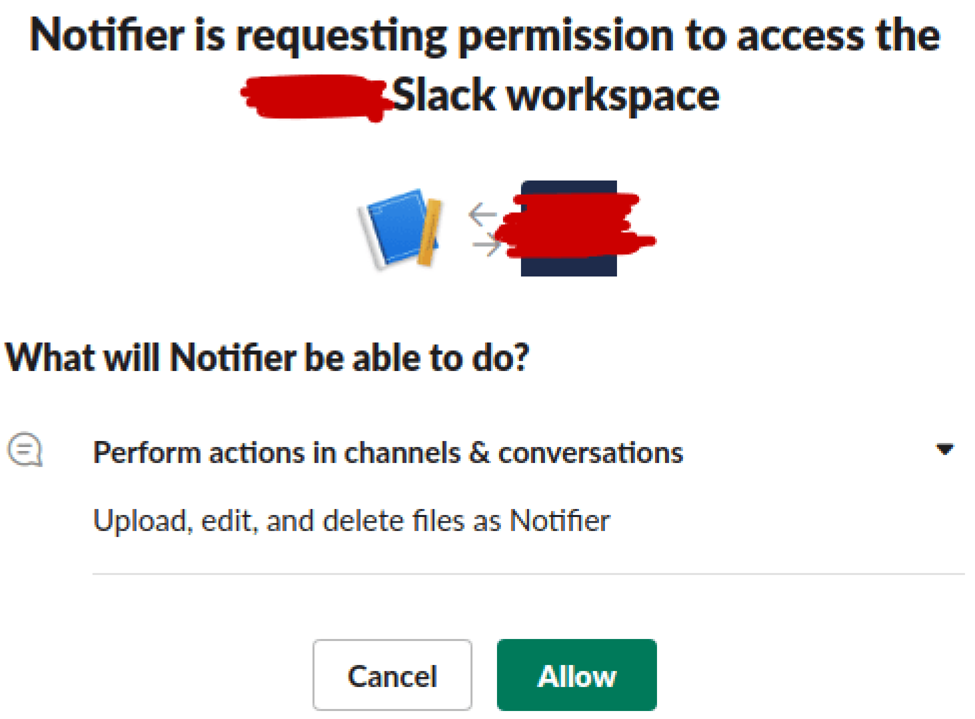
Bot Token Scopes

Scopes that govern what your app can access.

OAuth Scope	Description
files:write	Upload, edit, and delete files as Notifier
incoming-webhook	Post messages to specific channels in Slack

- Scroll up and click the "Install App to Workspace" button.

You can see what permissions are given to the app. Because I've added the "files:write" scope, I'm allowing the app to upload, edit and delete files.


Notifier is requesting permission to access the [redacted] Slack workspace

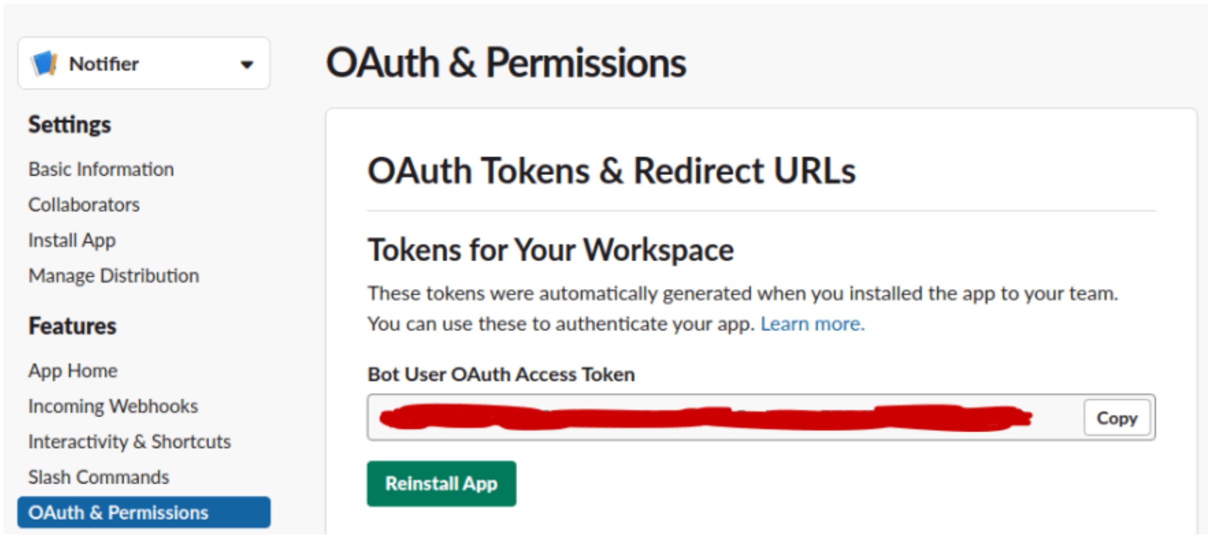
What will Notifier be able to do?

Perform actions in channels & conversations

Upload, edit, and delete files as Notifier

Click the "Allow" button.

- Copy and save the "Bot User OAuth Access Token".



Notifier

Settings

Basic Information
Collaborators
Install App
Manage Distribution

Features

App Home
Incoming Webhooks
Interactivity & Shortcuts
Slash Commands
OAuth & Permissions

OAuth & Permissions

OAuth Tokens & Redirect URLs

Tokens for Your Workspace

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

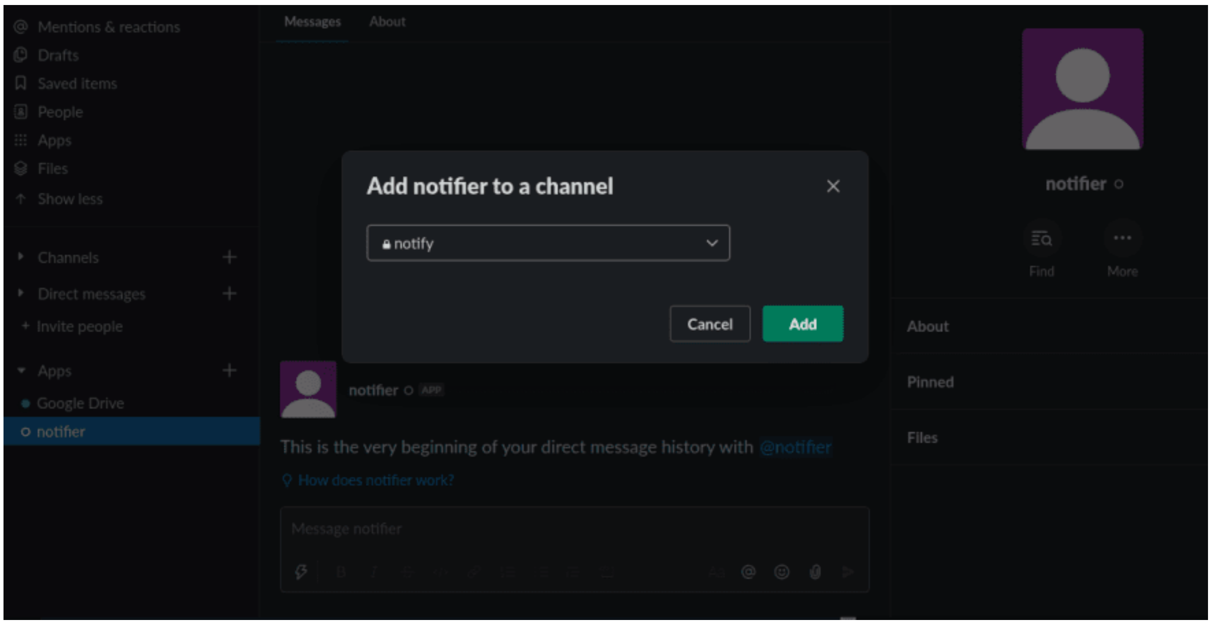
Bot User OAuth Access Token

[redacted] Copy

Reinstall App

Add the app to a channel

- Open the Slack app. In the Apps section click the bot name we've created. In the "Details" section, click the "More" button, then click the "Add this app to a channel...". Choose a channel and click the "Add" button.



Mentions & reactions
Drafts
Saved items
People
Apps
Files
Show less
Channels
Direct messages
Invite people
Apps
Google Drive
notifier

Messages About

Add notifier to a channel

notify

Cancel Add

notifier APP

This is the very beginning of your direct message history with @notifier

How does notifier work?

Message notifier

notifier

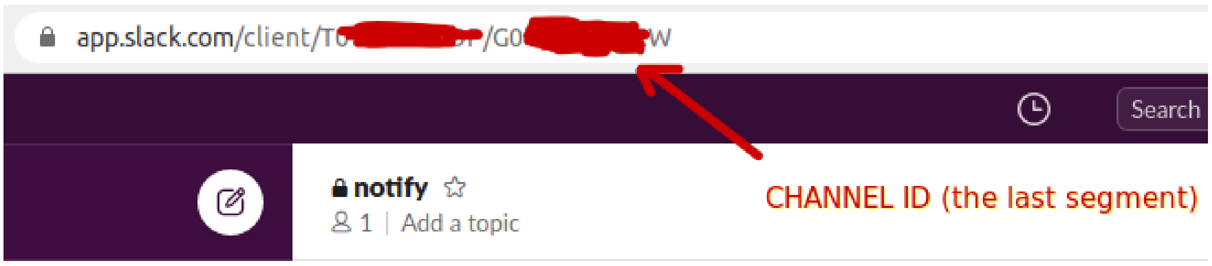
Find More

About
Pinned
Files

File uploading via API

- Before doing a call to the Slack API, we need to grab the ID of the channel we want to use to receive files. I couldn't figure out a better way to grab the channel ID, so if you know a better way - please share me a comment about that.

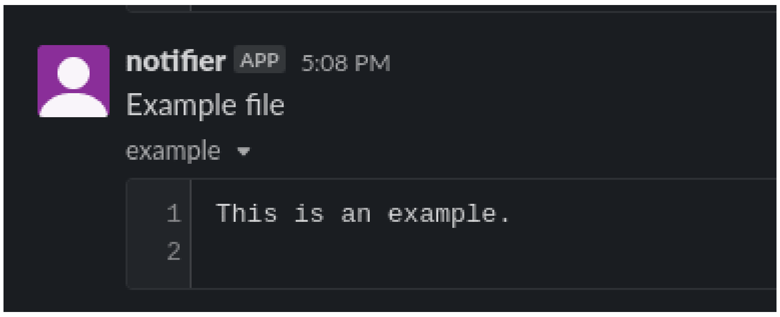
I simply used the web interface of the Slack app. URL contains a channel ID. Just choose the channel you want to use and grab the ID from URL (the last segment). It's something like `G1258QGKLMN`.


app.slack.com/client/T[redacted]/G0[redacted]W

- To test the file upload via slack API, we're going to use the `files.upload` endpoint. I'm using "curl" to do the call:

```
curl -F file=@example.txt -F "initial_comment=Example file" -F
channels=<CHANNEL_ID>, <ANOTHER_CHANNEL_ID_IF_NEEDED> -H "Authorization:
Bearer <BOT_USER_OAUTH_ACCESS_TOKEN>" https://slack.com/api/files.upload
```

It makes an attempt to upload the "example.txt" file to specified channels, and will add the "Example file" string to the actual slack message.


notifier APP 5:08 PM

Example file

example

```
This is an example.
```

Updating the `run-zap.sh` bash script from my previous posts

I just need to add an appropriate curl call to the bash script, and it should be enough to have the report shared to my Slack channel.
Here is the updated gist:
If you'll use that script and will need to have slack notifications working, do not forget to replace placeholders with real channel ID and Bot User OAuth Access Token as a Bearer token. If you don't need slack notification - simply comment out that line.
The usage of the script is simple:

```
bash run-zap.sh http://example.org h4x0r X-Corp TopSecret
```

So, the `http://example.org` parameter is the target host, "h4x0r" is the person/organization who prepared the report, and "X-Corp" is the one for whom the report has been prepared. And of course, the "TopSecret" is the project name. These extra parameters are used for the metadata of the report.