Workshop lab manual

⚡ PowerUp Red Teams Ops with RedELK ⚡

x33fcon, May 2023

Introduction to the workshop

This document provides guidance on the lab assignments of the workshop.

⚡ PowerUp Red Teams Ops with RedELK ⚡

This workshop aims for you to learn and experience:

- Spotting blue teams investigating red teaming infrastructure and gain better operational oversight with RedELK.
- Becoming more in control of your red team operations.
- For blue teamers, this will help you understand the artefacts that common investigation techniques leave behind.

Using a series of assignments, you will go from understanding, installing and configuring RedELK to maximizing its functionality for operational oversight and for detection of blue team activities.

We have prepared a realistic lab environment for you. This lab includes both **Outflank's OST Stage1 C2 and Cobalt Strike**!

For each lab we state a walkthrough, including screenshots and background information. It should give you enough guidance.
If you are stuck, do not hesitate to ask the trainers or your neighbor.

We hope you enjoy the ride ;-)

Introduction to your lab environment

Important note on Export Controlled goods Stage1 C2 and Cobalt Strike
To give you a representative lab environment, Stage1 C2 and Cobalt Strike (payload, server and client) are made available to you.

Stage1 C2 and Cobalt Strike are considered 'Intrusion Software' under the international *Wassenaar Arrangement on Export Controls for Conventional Arms and Dual-Use Goods and Technologies.* This means that access is subject to Export Controls.

Under no circumstances you are allowed to copy or otherwise copy software from your lab environment.

Access is monitored. Violation will result in immediate termination of your lab and future denial of access to these tools.

You are given a lab number in the form of two digits, e.g. 08. You will need to replace XX with your number in lab system names and accounts:

Host	URL or credential
Hostnames	
Accessing URL for Guacamole	https://access.redelk.rocks/
Labmaster (windows)	sXXlabmaster.redelk.lab
Target machine (windows)	sXXw10cli01.redelk.lab
C2 server	sXXc2.redelk.lab
Redirector	sXXredir.redelk.lab (internal) sXXredir.redelk.rocks (external, with CNAMEs): sXX-cs.redelk.rocks and sXX-s1.redelk.rocks.
RedELK server	sXXredelk.redelk.lab
Shared domain controller	vmdc01.redelk.lab

Credentials	
Guacamole and domain internal	studentXX@redelk.lab, password provided via Slack
Cobalt strike client	Username: free to choose Password: redelk
Stage1 web interface	Username: free to choose Password: shared separately to you

1. Lab 1 – Accessing your lab

This lab is all about getting setup with your lab environment.
We will do:

1. Registration and validations of name and email with workshop trainers
2. Make sure you can connect to your lab
3. Make sure you have a working C2 setup

1.1. *Registration and validation*

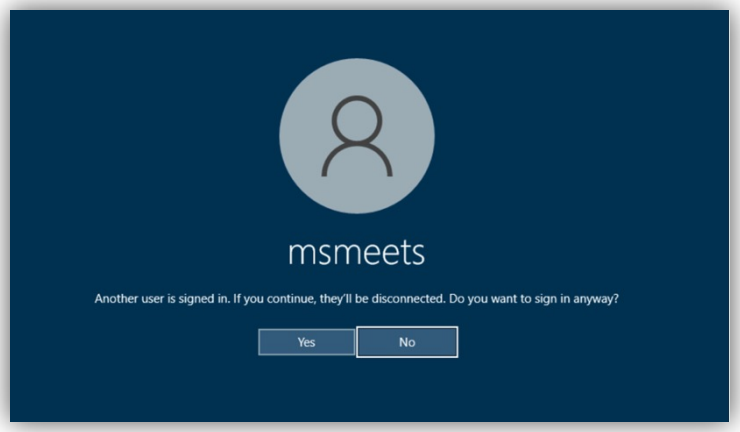
Please use the following steps to make sure you get access to our lab environment.

- You send an email with your full name and nationality to both trainers (marc.smeets@fortra.com and lorenzo@bernardi.be).
- Come see us with your valid government issues ID. We will validate your identity.
- We will invite that email address to the Slack workspace. Monitor your email inbox for an email from Slack. Connect to the workspace. You should see a public channel (all workshop attendees) and a private channel (only you and the trainers).
- You will receive lab access credentials in your private Slack channel.
- You verify access to your lab. Please let us know via Slack if you can get in.

1.2. *Connecting to your lab environment*

Now proceed with the following to see if you can connect to the lab. A few things to note:

- Connect to https://access.redelk.rocks. It is advised to:
  - Use a chromium based browser
  - Accept that the website can access your copy-paste buffer.
- Use the credentials you received via Slack to logon.
- In the left menu you should see a total of 5 systems you can connect to. Try them all ☺
- Get familiar with the Guacamole interface if you aren't already. For example use Control-Shift-Command (Mac) / Ctrl-Alt-Shift (Windows) for a menu to input methods, zoom settings and some other items.
- In some cases, a Windows machine you are connecting to may ask you if you want to log out another user (see picture below). That is ok to do, so click 'yes', and wait a few seconds.

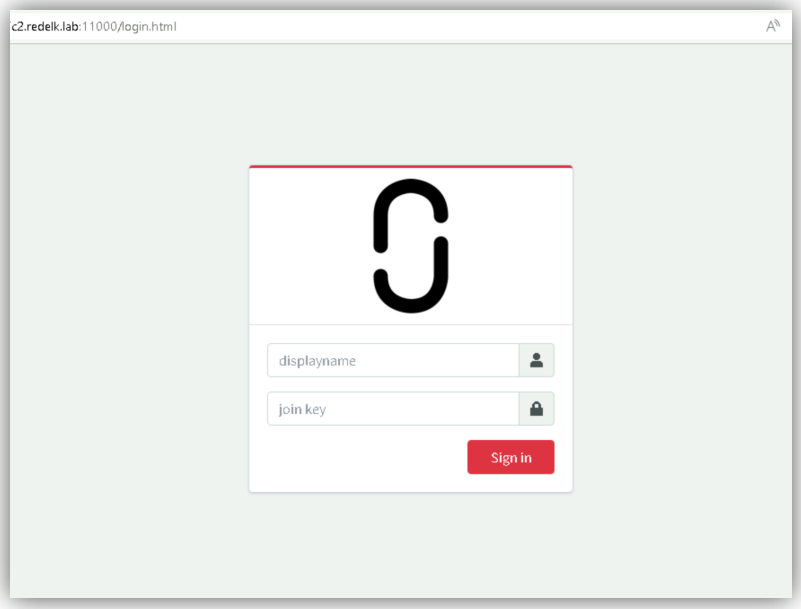


1.3. *Working C2 setup*

Stage 1 C2

Stage 1 C2 is preinstalled on your C2 server. All we need to do is provide it with a configuration file. This is done from your browser on the labmaster.

A. Start the browser on the labmaster VM and navigate to http://sXXc2.redelk.lab:11000. You will be presented with a logon screen similar to the one below.



B. Logon with a username of your liking, and the Stage1 password that was shared with you. You should be presented with the main screen of Stage1.
C. The only thing left to do is start a payload. You can find it on your labmaster in the `Desktop/tools` directory. Double-click it. You should see a new implant appear in the Stage1 console.
D. Click on the implant on the main screen. In a new screen you can give it commands. Start with help to get an idea of the Stage1 capabilities.

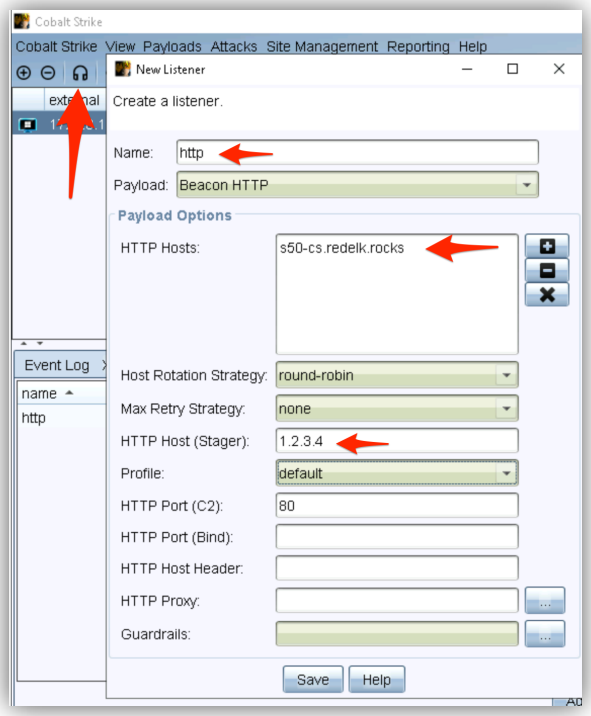
Cobalt Strike

First start the Cobalt Strike team server by opening the C2-SSH system in Guacamole.

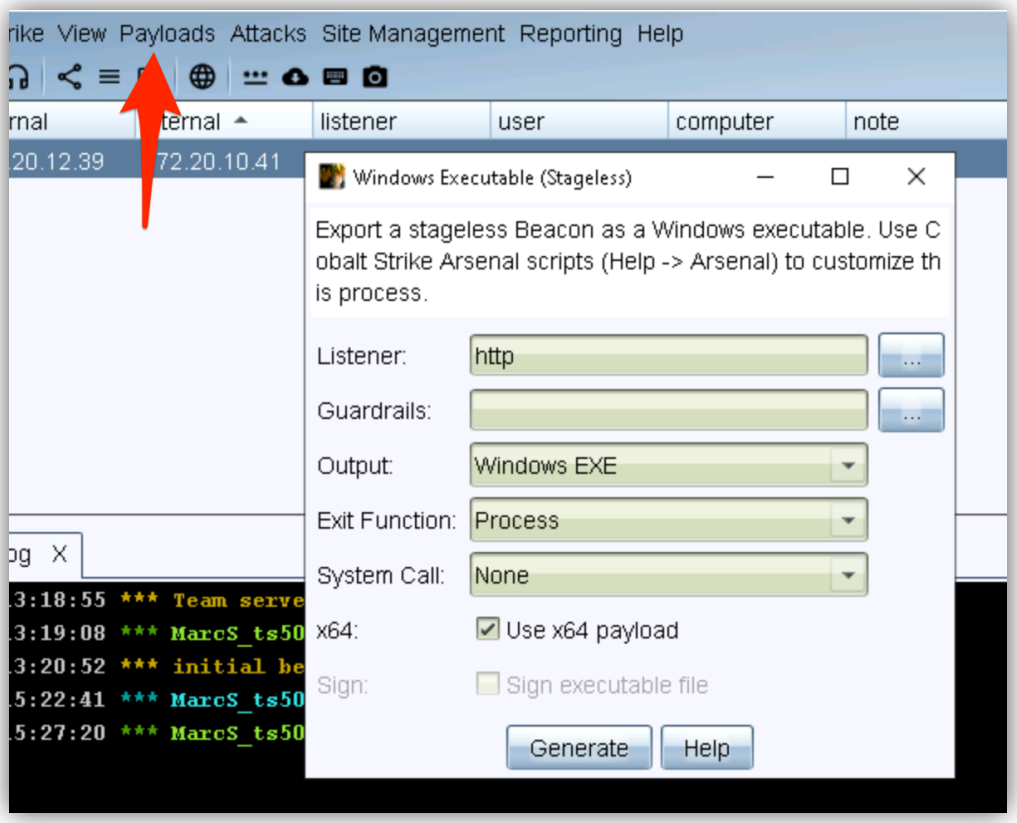
E. First, `sudo su -`
F. Navigate to `/root/cobaltstrike`
G. Execute the following command to start the cobalt strike server in a screen session, replacing the **C2IP** placeholder with the IP address of your C2 server:
   `screen -dmS teamserver ./teamserver **C2IP** redelk profiles/CS_redelk_workshop.profile`
H. Verify the teamserver is running by running `screen -l` or by resuming the screen session with `screen -r teamserver`.

Once the team server is running, start the Cobalt Strike client. You can find the Cobalt Strike client on the Labmaster `Desktop/Tools/cobaltstrike` folder.

I. Just double-click the `cobaltstrike.exe` file and it should start.
J. In the logon screen connect to:
   - Server: `sXXc2.redelk.lab`
   - Port: `50050`
   - User: `PickYourOwnUserName`
   - Password: `redelk`
K. Accept the certificate presented and make sure you are in the main screen.
L. Next step is to create a listener interface. Click on the microphone icon. In the popup enter the details as in the screenshot below, using your own lab ID. Pay attention to the HTTP Hosts with the `.rocks` instead of `.lab` suffix. Click `Save`.



M. Final step is to create a payload. We do this from the Payload menu. Click `Payload -> Windows stageless executable`, and enter the information as in the following screenshot.



N. Click `Generate` and save the output to the disk of the Labmaster, for example on the desktop.
O. Launch the Cobalt Strike beacon by double clicking it. You should now see a new implant appear in the Cobalt Strike window.
P. Right-click on the implant in the top window, and click `Interact`.
Q. In the newly presented tab you can interact with the implant. Type `help` to get an idea about the Cobalt Strike capabilities.

2. Lab 2 – RedELK setup

This lab is all about setting up RedELK
We will do:

1. Install RedELK on your infra.
2. Validate if data is ingested.

It is smart to also take a look at the RedELK wiki on github, besides this lab manual. The wiki may have more information on installation and debugging. Make sure to check the following pages:

- https://github.com/outflanknl/RedELK/wiki/Naming-requirements-within-RedELK
- https://github.com/outflanknl/RedELK/wiki/Generating-keys-and-packages
- https://github.com/outflanknl/RedELK/wiki/RedELK-server-installation
- https://github.com/outflanknl/RedELK/wiki/Redirector-installation
- https://github.com/outflanknl/RedELK/wiki/C2-server-installation

2.1. *Install RedELK*

Key generation

First, we need do the initial install step. This will generate relevant installation packages for all infra components.

A. On the RedELK server make sure you are running as root and clone the RedELK repo:

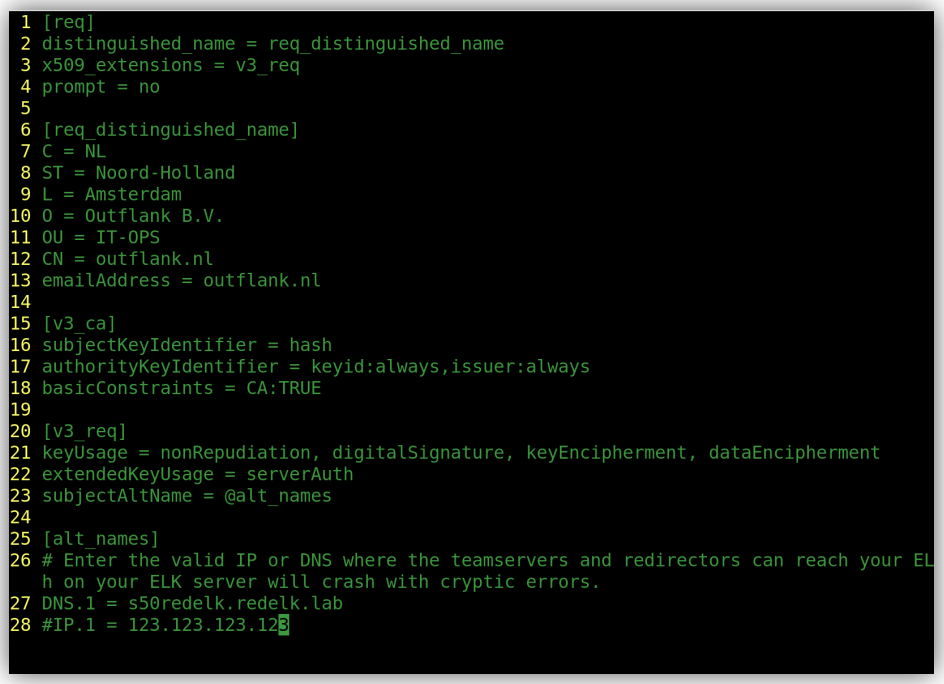
```
# On the RedELK server
# Become root
sudo su -

# clone the repo
git clone https://github.com/outflanknl/RedELK.git

# go to the new directory
cd RedELK

# edit the certificate file
cd certs
cp config.cnf.example config.cnf
vi config.cnf # or any other editor you prefer.
```

Make sure to edit the file correctly. Pay special attention to the IP and DNS statements at the end of the file. Enter the correct DNS name where your redirector and C2 server will reach your RedELK server on. An example is shown in the picture below.



```
[req]
distinguished_name = req_distinguished_name
x509_extensions = v3_req
prompt = no

[req_distinguished_name]
C = NL
ST = Noord-Holland
L = Amsterdam
O = Outflank B.V.
OU = IT-OPS
CN = outflank.nl
emailAddress = outflank.nl

[v3_ca]
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid:always,issuer:always
basicConstraints = CA:TRUE

[v3_req]
keyUsage = nonRepudiation, digitalSignature, keyEncipherment, dataEncipherment
extendedKeyUsage = serverAuth
subjectAltName = @alt_names

[alt_names]
# Enter the valid IP or DNS where the teamservers and redirectors can reach your EL
h on your ELK server will crash with cryptic errors.
DNS.1 = s50redelk.redelk.lab
#IP.1 = 123.123.123.123
```

B. Once done with editing, it is time for the actual initial setup.

```
# On the RedELK server in the /root/RedELK directory
./initial-setup.sh certs/config.cnf
```

You will have 10 seconds to verify the correct IP/DNS info. If incorrect, just abort with CTRL-C.

```
 ____          _ _____ _     _  __
|  _ \ ___  __| | ____| |   | |/ /
| |_) / _ \/ _` |  _| | |   | ' /
|  _ <  __/ (_| | |___| |___| . \
|_| \_\___|\__,_|_____|_____|_|\_\


This script will generate necessary keys and packages for RedELK deployments


[*] Will generate TLS certificates for the following DNS names and/or IP addresses:
DNS.1 = s50redelk.redelk.lab

[!] Make sure your ELK server will be reachable on these DNS names or IP addresses or your
[*] Abort within 10 seconds to correct if needed.
[*] Creating certs dir if necessary
[*] Creating config files form example files
[*] Generating private key for CA
[*] Creating Certificate Authority
[*] Generating private key for ELK server
[*] Generating certificate for ELK server
[*] Signing certificate of ELK server with our new CA
[*] Converting ELK server private key to PKCS8 format
[*] Copying certificates to relevant redir and c2servers folders.
[*] Copying certificates to elkserver directory.
[*] Creating ssh directories if necessary
[*] Generating SSH key pair for scponly user
[*] Copying sshkeys to relevant folders.
[*] Copying VERSION file to subfolders.
[*] Creating TGZ packages for easy distribution


[*] Done with initial setup.
[*] Copy the redirs.tgz, c2servers.tgz and elkserver.tgz packages to every redirector, c2se
there locally.
```

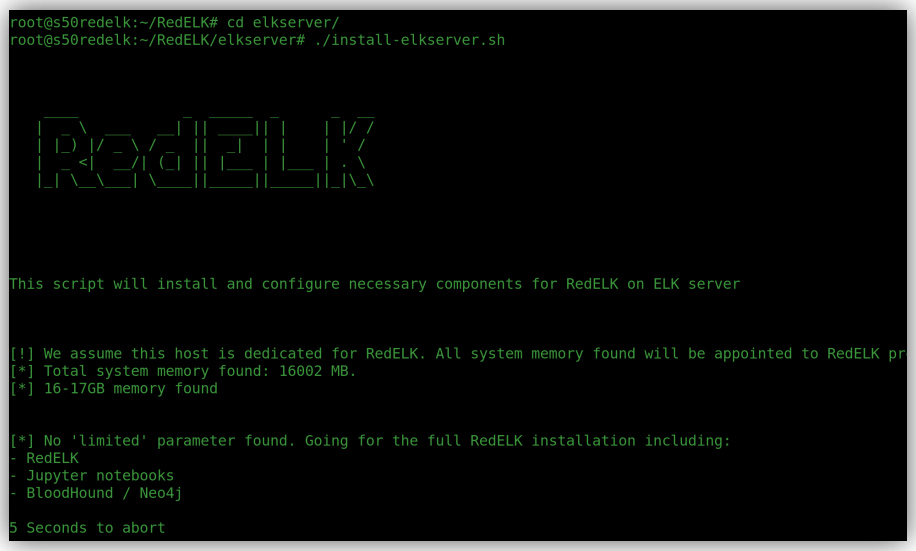
RedELK server installation

C. The installation of the core RedELK server is rather simple. Just run the installer, optionally with parameters.

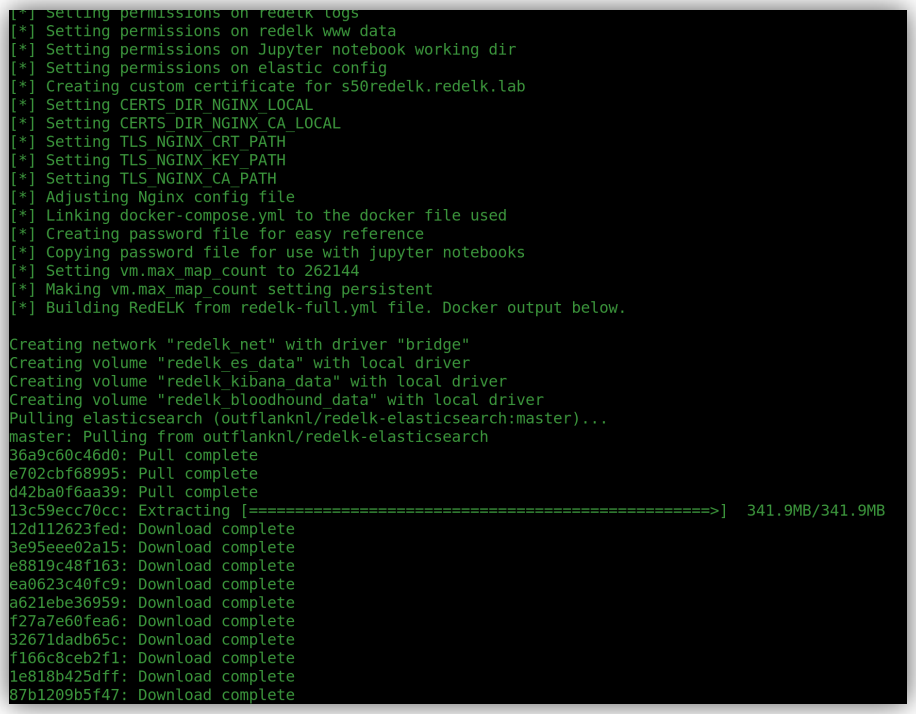
```
# On the RedELK server, change to the elkserver directory
cd elkserver

# Run the installer
./install-elkserver.sh
```

Output at start will look like this:



Output during the installation will look like this:

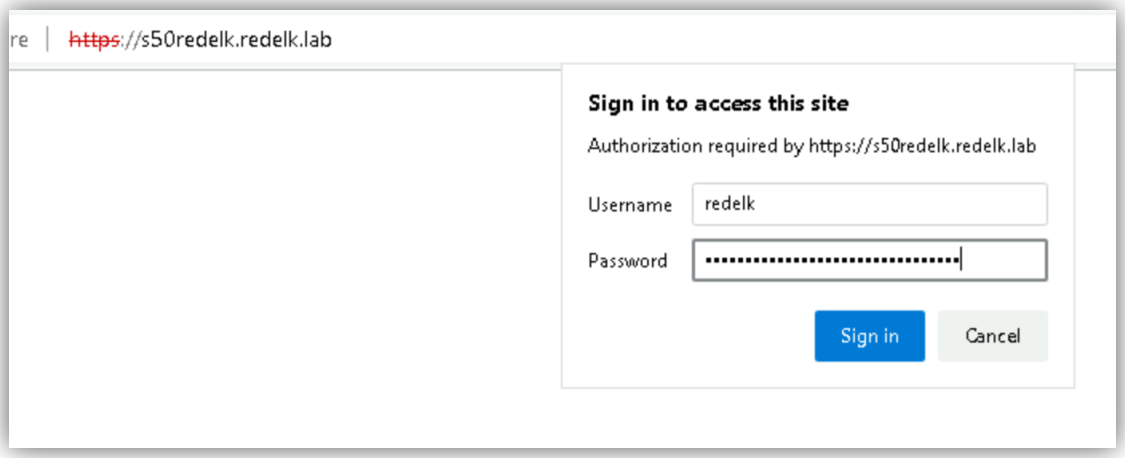


Once done with the installation it will report the generated passwords to the screen. They are also stored in the `redelk_passwords.cfg` file. Please note this is an output file, not a config file. Should you want to change the password you should edit the `.env` file and restart the docker containers.

D. Next step is to setup file syncing between the RedELK and C2 servers. This is configured in the file `mounts/redelk-config/etc/cron.d/redelk`. Modify the line mentioning the "`getremotelogs.sh`" / rsync line and add the details of your c2 server.

Pay attention to the name you give to your c2server in this file. It Should match the name parameter you provide during the installation on the c2server later on. In our case it will be `s50c2` for lab 50.

E. Logon tot the Kibana web interface with the `redelk` user account using the password that was saved to the `.env/redelk_passwords.cfg` file.



You can ignore errors on `missing indices` as the indices will be created later on when data arrives in the ELK stack.
You can also ignore the error on a missing `server.PublicBaseUrl`. We don't really care about that.

Redirector installation

Now install the redirector component of RedELK.

F. Secure copy the redirs.tgz file to your redirector using the student account.

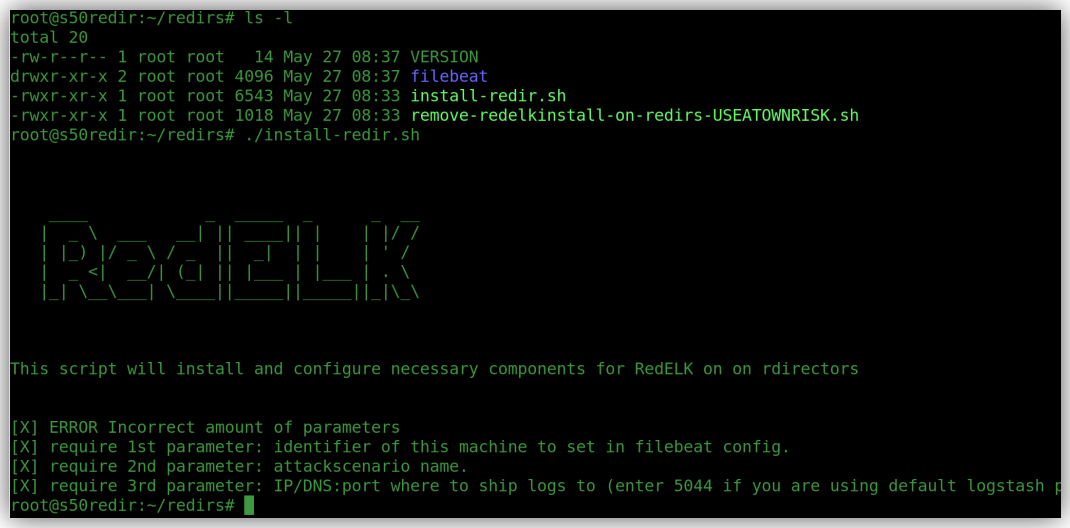
```
# On the RedELK server, scp the redirector installation file
scp /root/RedELK/redirs.tgz studentXX@sXXredir.redelk.lab:
```

G. Now open a Guacamole session to the redir. On the redir move the redirs.tgz file to the /root folder and extract it..

```
# On the redirector
sudo su -
mv /home/studentXX/redirs.tgz /root
```

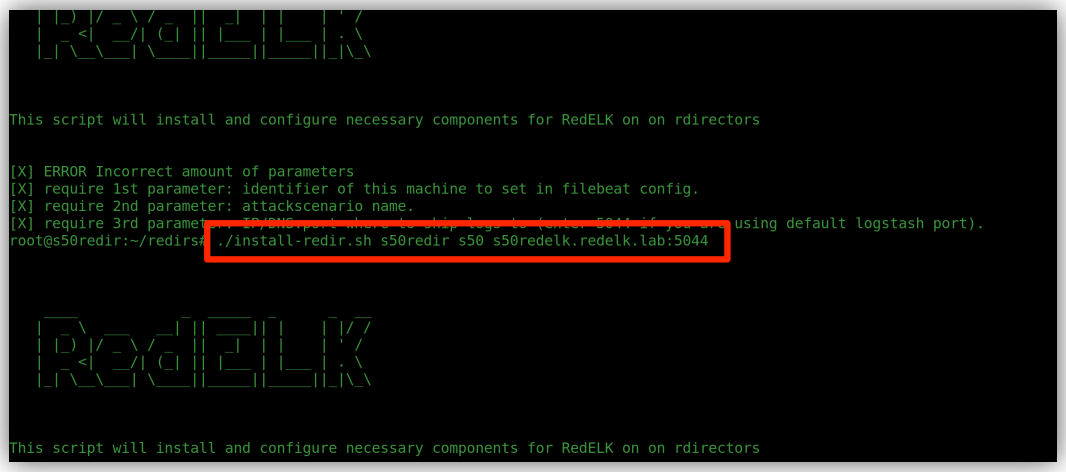
```
cd /root
tar zxvf redirs.tgz
cd redirs
```

H. Run the installer without any parameters to better understand what is required



I. Do the actual installation using:
   a. Parameter 1: identifier of this machine as it will appear in the Kibana interface, e.g. `sXXredir`
   b. Parameter 2: your scenario name as it will appear in the Kibana interface, e.g. `sXX`
   c. Paremeter 3: the url and port of your RedELK installation, e.g. `sXXredelk.redelk.lab:5044`

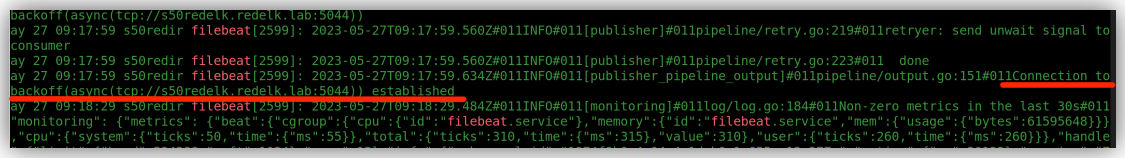
```
# On the redirector, run the installer
./install-redir.sh sXXredir sXX sXXredelk.redelk.lab:5044
```



J. After a few seconds check if filebeat is working.

```
# On the redirector
# check the service status
service filebeat status
```

```
# check for 'established' connections in the filebeat log
grep filebeat /var/log/syslog
```



C2 server installation

Now install the c2server component of RedELK.

K. Secure copy the c2servers.tgz file to your c2 server using the student account.

```
# On the RedELK server, scp the c2server installation file
scp /root/RedELK/c2servers.tgz studentXX@sXXc2.redelk.lab:
```

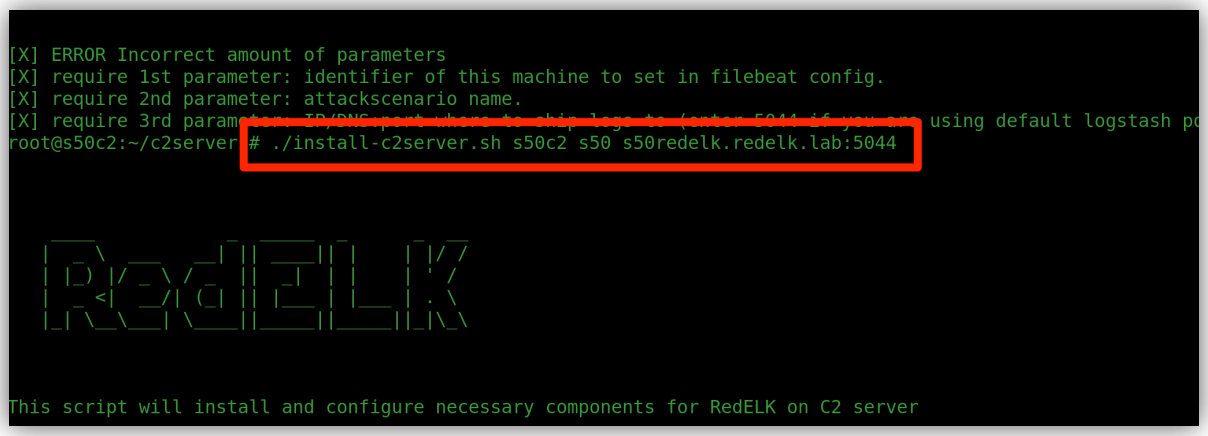
L. On the c2server move the c2servers.tgz file to the /root folder and extract

```
# On the c2server
sudo su -
mv /home/studentXX/c2servers.tgz /root
cd /root
tar zxvf c2servers.tgz
cd c2servers
```

M. Run the installer without any parameters to better understand what is required.

N. Do the actual installation using:
   a. Parameter 1: identifier of this machine as it will appear in the Kibana interface, e.g. `sXXc2`
   b. Parameter 2: your scenario name as it will appear in the Kibana interface, e.g. `sXX`
   c. Paremeter 3: the url and port of your RedELK installation, e.g. `sXXredelk.redelk.lab:5044`

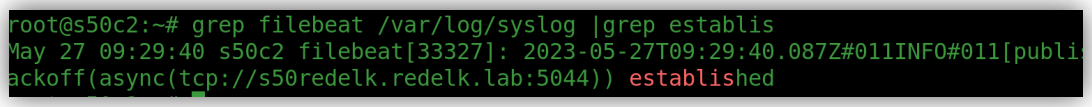
```
# On the c2server, run the installer
./install-c2server.sh sXXc2 sXX sXXredelk.redelk.lab:5044
```



O. After a few seconds check if filebeat is working.

```
# On the c2server
# check the service status
service filebeat status

# check for 'established' connections in the filebeat log
grep filebeat /var/log/syslog
```



2.2. *Validate data is ingested*

P. Return to the Kibana web interface and verify if data is present there. It should contain data from both the C2 server (`rtops-` index) and the redirector (`redirtraffic-` index). The summary dashboard may still produce some errors and that is OK. You most likely have not yet gathered credentials, screenshots, etc so it complains some data can't be shown just yet.

Q. If both indices have data from their respective host, you are good to go. If not, it is time for troubleshooting.

R. Also check if the rsync of implant logs works by clicking on an implant url in the rtops index. It should take you to a text file containing the full raw impant log.

Troubleshooting

If you have insufficient or no data at all in the Kibana web interface, you can try the following to locate the issue:

- A common issue is incorrect TLS certificate information entered in the initial setup. If this is the case, the filebeat on the sides of the C2 and redir will report in their logs that they 'backoff' instead of 'connection established'. You have two ways of fixing:
    - Quick 'n dirty: disable tls encryption on the sides of filebeat and logstash in their respective configuration files.

    - Edit the certificate config file, remove old certificate files, regenerate tls certificates and copy them over to the right locations. The RedELK sever requires all new files. The redir and c2serevr only require a selection of files as you see in the respective filebeat directories and configuration files.
- Check the installer log files on the 3 components for indications.
- Use the debugging steps listed on the wiki page for each specific component:
    - https://github.com/outflanknl/RedELK/wiki/RedELK-server-installation#debugging
    - https://github.com/outflanknl/RedELK/wiki/Redirector-installation#debugging
    - https://github.com/outflanknl/RedELK/wiki/C2-server-installation#debugging

3. Lab 3 – Operational oversight

This lab is all about using RedELK for operational oversight.
We will do:

1. Generate data by running multiple implants from multiple lab systems
2. Use RedELK to get insight into the dat

3.1. *Run implants*

This step is rather free format. Just make sure that you generate enough logs.
Some things to think about:

- Run a small webscan on the redirector's external interface to get some more traffic logs.
- In a existing Cobalt Strike implant:
  - make a screenshot
  - download a file
  - run a keylogger
  - perform a portscan to your target machine: portscan `sXXw10cli01.redelk.lab`
- Start an administrative beacon of Cobalt Strike on your labmaster `(right-click, run as admin.`
- In an administrative beacon, run `hashdump` to get credentials of the local system.
- Do lateral movement to your target machine. This can be done manually with the `jump` command, or you can click `Targets` on the top bar, select your target machine, and select a lateral movement technique. You can use the student credentials, but change the domain realm to `redelk`, and only use an administrative session to jump from.
- On the new beacon, also run a few commands to get some extra log data.

3.2. *Getting insight with RedELK*

This step is rather free format. Just click around in the RedELK Kibana interface and get yourself accustomed to the interface. Some things to think about:

- Do a hard browser refresh in the Kibana screen. This will make sure that your browser understands the new fields that become available.
- Although RedELK's initial view in the Kibana app showing summaries, it might be useful to start with the Discovery screens and explore the different pre-made vies there.
- Open the Red Team Traffic View and explore some events, check the enrichment that is performed in the background and do some querying using the GUI buttons or the query language.
- Open the Red Team Operations View and explore some events. Look at the enrichment and do some querying.
- Explore the other Views available.
- Explore screenshots, it should have thumbnails and the full screenshots
- Explore Downloads and keystrokes.
- Go back to the RedELK app and explore the different pages with dashboards.

4. Lab 4 – Advanced configuration of RedELK

This lab is all about advanced configuration of RedELK.
We will do:

1. General config
2. Configure API keys for VT, IBM X-Force or Hybrid Analyses
3. Configure Slack or Teams web hook
4. Configure email
5. Run test alarm
6. IP and other list files

The RedELK wiki has lots of details about configuration at https://github.com/outflanknl/RedELK/wiki/RedELK-server-installation#configuration

4.1. *General config*

A. Get familiar with the main config file in on the RedELK server in `mounts/redelk-config/etc/redelk/config.json`.
B. Change at least the `project_name` setting to something that is more specific than the default `redelk-project`. This will be included into every alarm you receive. So especially when you have multiple operations running in parallel, you want this to immediately let you know what project the alarm applies to.

4.2. *Configure API keys for VT, IBM X-Force or Hybrid Analyses*

To alarm you about your artefacts known by public scanning engines, RedELK supports checking at three major engines. You will need to provide an API key to RedELK in order to check with these scanning engines.

They will require account creation in order to get an API key. More info at:

A. https://support.virustotal.com/hc/en-us/articles/115002088769-Please-give-me-an-API-key
B. https://www.ibm.com/docs/en/qns/5.4.0?topic=integration-obtaining-api-key-password
C. https://www.hybrid-analysis.com/docs/api/v2
D. Get an API key at one or more of these services, at least Virus Total or IBM X-Force.
E. Enter the API keys in the RedELK config file on the RedELK server in `mounts/redelk-config/etc/redelk/config.json`, section `alarm_filehash`.
F. Enable the checks by setting the `enabled` parameters to `true`.

IBM X-Force and Virus Total can also be used for getting domain classifications. This is done by:

G. Telling RedELK which domains it should check by listing them in `mount/redelk-config/etc/redelk/domainslist_redteam.conf`.
H. Enable this type of check and give the API key in the `enrich_domainscategorization` setting.

Results of these checks are put in the bluecheck index. Right now, alarms on this type of info is still in development. For example, changes of the domain classification and on rogue classification altogether..

4.3. *Configure Slack or Teams web hook*

To get alarmed by RedELK via Slack or Teams you need to give it a web hook URL. More info at generating these can be found here:

- https://learn.microsoft.com/en-us/microsoftteams/platform/webhooks-and-connectors/how-to/add-incoming-webhook?tabs=dotnet#create-an-incoming-webhook-1
- https://api.slack.com/messaging/webhooks

I. If you have got a webhook, enter it in the config file and enable the notification.

4.4. *Configure email*

If you want to get alarms from RedELK via plain old email you need to give it the SMPT settings in the `notifications.email` section.

J. If you have these settings available, enter them here. If not, make sure to use a webhook or you will not receive any alarms.

4.5. *Run test alarm*

To verify if your alarm setup is running, generate a test alarm. This is done by:

K. Enabling `alarm_manual` in the config file

L. Type something in the Cobalt Strike eventlog that starts with `REDELK_ALARM` and is followed by the message you want in your alarm.

4.6. *IP and other list files*

A second set of configuration is used for alarming about specific IP addresses or user agents you want to be alarmed about. This is all defined in the files in `mount/redelk-config/etc/redelk/`. These alarms can be enabled or disabled in the main config file.

M. If you still have some time left, feel free to look at the content of these files, the wiki to get an idea of what these do and what these alarms actually do in the background by looking at the actual alarm modules in `docker/redelk-base/redelkinstalldata/scripts/modules` on the RedELK server

5. Lab 5 – Detecting blue team activity

This lab is all about detecting blue team activity.
We will do:

1. Detection of publicly known malware
2. Detecting investigative traffic to your infrastructure

5.1. *Detection of malware investigation*

To get notified about blue teams uploading files of your operation to sandboxes, RedELK has a check for you. We can simulate this in the workshop.

A. Enable `alarm_filehash` in the config
B. Download a version of Mimikatz.exe (for example here: https://github.com/ParrotSec/mimikatz/blob/master/x64/mimikatz.exe) to your labmaster.
C. Upload Mimikatz.exe to one of your targets via Cobalt Strike, Cobalt Strike will report an IOC and RedELK will pick this up to validate at public scanning engines.

A few minutes later you should get an alarm from RedELK stating your IOC was found and give some detailed info about the scanning results.In case you do not get the alarm, there is a few things you can do to troubleshoot:

- Check if RedELK has recognized it as an IOC by looking at the IOC overview in Kibana
- Check the status of the alarm by looking at the daemon log in `mounts/redelk-logs/daemon.log`

5.2. *Detection of investigative traffic*

To get notified about blue teams investigating your infrastructure, RedELK has several alarms for you. We can simulate this in the workshop.

D. Make sure to enable the alarms `alarm_httptraffic` and `alarm_useragent` in the config file.
E. To trigger the alarm, send a web request to your redirector that mimics that of your implant. In your lab this is http://sXXredir.redelk.rocks/s1 or http://sXXredir.redelk.rocks/cs - you can validate in your HAProxy config on the redirector.
   You can manually trigger this using curl from your local system. For example:
   `curl -v http://sXXredir.redelk.rocks/s1/fdsfds`

This should trigger alarms as this 1) contains the curl user agent that RedELK alarms you about as configured in the file `mount/redelk-config/etc/redelk/rogue_useragents.conf`, and 2) it is coming from an IP address that is not whitelisted in `mount/redelk-config/etc/redelk/ iplist_redteam.conf`.

6. Lab 6 – Advanced topics

We have no set walkthrough for this. This lab is all about what you make of it ☺

A few things you can consider doing here:

1. Try to trigger the other alarms RedELK has.
2. Add your own C2 or custom alarm to RedELK.
3. Play with Stage 1 or Cobalt Strike.
4. Help with open GitHub issues and improvements.

No walkthrough here, you are all on your own.