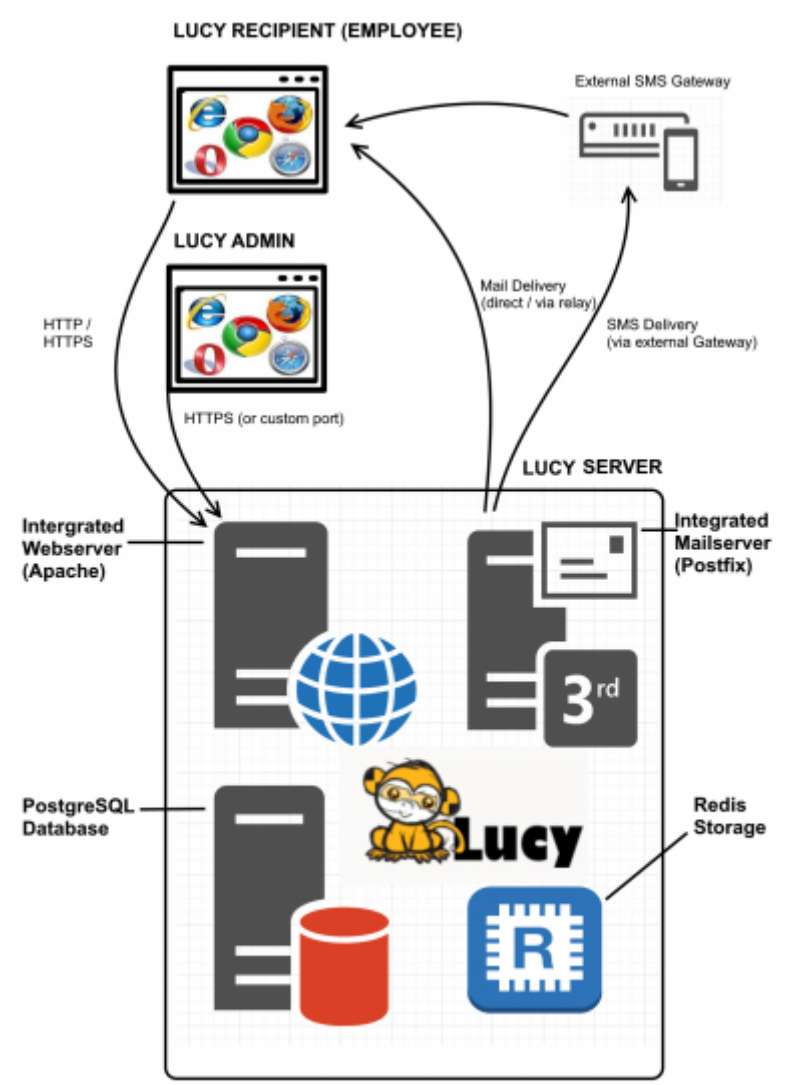
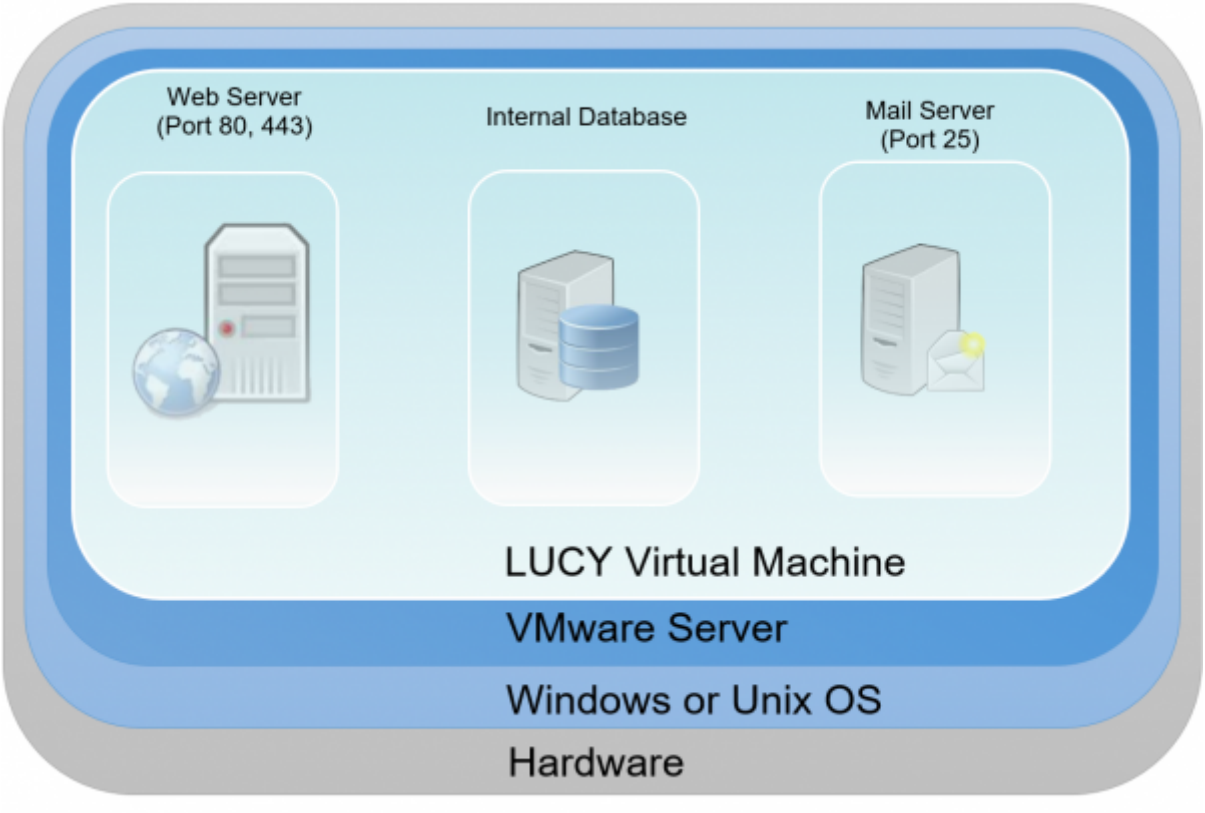
Technical Information

This page describes the technologies and software used by LUCY. The information herein applies mostly to VM-based LUCY distributions (VMware Workstation, VMware ESXi, VirtualBox and Amazon) as LUCY installations on custom dedicated servers may have different configurations. The LUCY server contains different components. Those components are already installed when using our images (Vmware/AMI etc.) or are installed when using the Linux installer.



LUCY Vmware technical components

When you download and boot the VMware Image, all software components are integrated in that image. There is no need to install any additional software. All components (DB, mail server, web server etc,) are bundles within the VMware images and controlled by the internal LUCY software, which runs transparently in the background. The updating of those components is also done within the LUCY software through internal processes, which are not visible to the end user.



Browser & OS support

The list of supported browser and OS can be found here.

Hardware requirements

The hardware requirements may vary depending on the amount of recipients you upload for testing, your campaign type and also how you run the campaign. More info here.

External service API's

LUCY offers different API's to third party services or via REST API:

- URL Shortening services
- DynDNS services
- External mail relays (Sendgrid)
- SMS (Messagebird)
- Domain (Godaddy)
- LUCY REST API

Installation

Lucy can be installed on premise or in the internet. More info here.

Network communication

LUCY may require certain communication channels to servers on the internet. More info here.

Operating System

Starting from 4.3, LUCY is running on a 64-bit **Debian 9 (Stretch)** system. There are no system patches or hardenings applied - LUCY uses a vanilla Debian distribution without any additions. The system is configured to download updates and new packages from a custom LUCY apt mirror, which has the same IP address, as LUCY license server (make sure it is open on your corporate firewall). The operating system gets updated only when you upgrade LUCY to a new version.

Security Settings

- Create custom firewall rules (iptables) in LUCY
- SSL Configuration
- Custom Admin URL & Title
- Confidentiality of Campaign Data
- Security Considerations
- Firewall & Security Settings
- Password Policies, Login Protection & Strong Authentication
- 2-Factor Authentication (2FA)

Accounts

There are a few accounts on the system like "phishing" or "support". The "phishing" account is the one that is used for Lucy file permissions. Also you can SSH to Lucy using "phishing" account and it will automatically launch the console setup program. The "phishing" account is required for Lucy to function properly, so it's not recommended for removal. If you SSH to the "phishing" account the console setup program (python setup script) is launched (with elevated privileges). So it can't be used as a full-featured SSH login. The purpose is to only run the setup console.

The "support" account is used to log into the system over SSH when user turns on SSH access for support. It's safe to remove the "support" account, but then we won't be able to log into that system over SSH for support purposes.

Web Server

Lucy web interface uses **Apache 2.4** as a web server. The server utilizes "mod-security" and "mod_headers" modules to hide underlying software signatures from external visitors.

Database

LUCY stores all related data in **PostgreSQL 9.6** RDBMS. All sensitive information stored in there is

encrypted as PostgreSQL is available only for internal connections. There are no configurable options for the DB encryption. The encryption is mandatory for all data and is performed automatically with the following settings:

- It’s a column-level encryption performed on both the application and DB layers before storing any data in the database. We don’t use TDE (transparent database encryption), as PostgreSQL doesn’t support it, so we encrypt only a subset of columns in DB – everything that holds client/attack/recipient-related data.
- We mostly perform the encryption/decryption on the application level, but there are certain queries that decrypt data on the DBMS level for convenience – for sorting & data search.
- The encryption is performed using AES-256-CBC.
- On demand we can provide a HSM solution, that will allow us to use a HSM-based encryption – in that case the encryption key will be stored on the external hardware module with anti-tampering protection.

Intermediary storage

LUCY also uses **Redis Server 3.2** as an intermediary storage (task queue) when passing input data and results between users and background system workers. The data stored in Redis is not encrypted.

Software, secure coding & hardening

LUCY code is a **PHP** application based on **Yii Framework**. Certain parts of the system (some background scripts) are implemented using the Python programming language. We follow the following security principles:

- OWASP Top 10 / 2017
- Framework-level SQL injection prevention
- Framework-level CSRF prevention
- Lucy partially conforms to "CIS Debian 8" checklist (50% conformance: we can provide a detailed list of non-conforming items upon request. There are no critical issues in uncovered parts)
- Lucy partially conforms to "CIS PostgreSQL 9.5" checklist (50% conformance: we can provide a detailed list of non-conforming items upon request. There are no critical issues in uncovered parts)

Folders

Almost all LUCY files are located in /opt/phishing folder, which contains the system code, user files and settings. Normally, you shouldn't have to deal with LUCY code, so the most useful directories are those where user files are stored. All user-related files are placed in /opt/phishing/files folder:

- Attachment-templates — File templates for file-based attacks (exes and macro files)
- Awareness-templates — Awareness template storage
- Campaigns — Campaign and scenario files storage
- Domains — Domain DKIM configuration storage
- Header-images — Header images for report templates
- Page-templates — System page templates (for example, 404 page template)

- Recipient-groups — Recipient group storage
- Report-templates — Report templates file storage
- Scenario-templates — Scenario templates file storage
- System — Keeps system files: Custom logo, system SSL certificates and system-wide static files for landing pages

Logs

There are several places in the system where you can find LUCY-related logs. They can be helpful to resolve or diagnose issues using your own technological resources without involving LUCY support. First of all, you should look at Apache web server logs directory /var/logs/apache2, where web server saves access and Error log files:

- access.log
- error.log

The next place with logs is /opt/phishing/runtime, where LUCY application logs are stored:

- application.log — Web application log - here you can find all web interface error notifications.
- beef.log — BeEF vulnerability detection framework log.
- console.log — Console commands log - Contains errors and issues for periodical background commands.
- resque_system.log — System background tasks log.
- resque_worker.log — Service background tasks log - Almost all background tasks you use in LUCY (including sending emails, copying websites, etc.) write logs to this file.
- scheduler.log — Scenario scheduler log.

Critical Services

There are several system services that are critical for LUCY. You can check if they are running by executing "ps aux" command. If some of required services are not running, then they should be started using "service NAME start" command (where NAME is the name of the service you are going to start):

- apache2
- postgres
- redis-server
- supervisor

Frequently asked questions (FAQ)

Which components are installed with the installer script? What modifications are made to Linux OS by the installer?

When Lucy is installed as a docker image, it gets downloaded from https://hub.docker.com (you can search for Lucysecurity there) to internal docker storage that is managed by docker itself – as any

other usual docker image. Usually it's **/var/lib/docker/**. After that the installer launches the image with specific parameters, extracts a management script (that launches Lucy's console setup tool) to **/tmp/files**, moves it to **/usr/bin/lucy-setup.sh** and removes the **/tmp/files** folder. These are all changes that the docker-based installer does, all other software and files are within the container.

When installing Lucy in a "legacy" way (which is possible only for Debians, as the installer requires some Debian-specific packages and software versions), the installer adds around 500 new packages, does very deep system changes, including configuring the login shell, changing other services (postfix, apache, etc).

The installer checks if docker is installed and installs it from the official docker website. Are there any downsides of having docker engine pre-installed?

No, pre-installed docker engine is just fine and will work completely the same. Just make sure it will be updated on time.

Who patches the Docker Runtime and host Linux system when security vulnerabilities become known?

When using a Docker-based installation, the Docker engine and all packages on the host are managed by the host package management system – be it RHEL package management or Debian/Ubuntu APT. Lucy's docker container has no access to outer system and therefore cannot install any updates there. When using a "legacy" installation, all software packages are updated along with Lucy updates using Lucy's Debian mirrors.

How does Lucy update when it's running in Docker?

After installing a Docker container with Lucy, Docker's container system is not involved anymore - all updates are performed within the container using container's APT system with our custom package mirror.

Is the container included in the installer payload or is the container loaded via docker pull?

Lucy's container is downloaded from https://hub.docker.com (official Docker Hub) during installation.

When configuring proxy in Lucy, what software is affected?

In case of Docker installation, only software inside the container is affected. There are no consequences for the software on the host system. When using a legacy installer, the proxy configured will be used as a default proxy for the whole Linux system.

How do you back up Lucy?

Speaking of Docker container installation, the backup is seamless and is a peace of cake. First of all you "commit" the container, making a static image of it with all files, changes, etc: **docker commit lucy lucy-backup** After that you:

- either save it as a tar file: **docker save -o /path/to/lucy-backup.tar lucy-backup**
- or export it to your private docker registry: **docker push lucy-backup**

The backup image can be recovered easily from file by **docker load -I /path/to/lucy-backup.tar** (in case of local file) or **docker pull lucy-backup** (in case of private docker registry)

After that you can start your container as a regular docker container with this command: **docker run –privileged -v /proc/sysrq-trigger:/sysrq -d -p 80:80 -p 443:443 -p 25:25 -p 5001:5001 –name lucy –restart=always lucy-backup /bin/bash /etc/run-services.sh**

Things get a bit more difficult in case of "legacy" installation - you should back up multiple directories, where configurations and files are stored:

- /opt/phishing
- /etc/
- /var/lib

From:
https://wiki.lucysecurity.com/ - **LUCY**

Permanent link:
https://wiki.lucysecurity.com/doku.php?id=technical_information&rev=1564051801

Last update: **2019/07/25 12:50**