Fitcrack - user manual

Radek Hranický
Faculty of Information Technology
Brno University of Technology

Contents

Fitcrack - user manual

I. Introduction

Fitcrack[1] is a distributed password cracking system based on BOINC platform, developed by *Fitcrack team* within *Networked and embedded systems research group*[2] (NES@FIT). As a main "cracking engine", Fitcrack uses **hashcat** tool, which performs password hash calculation on computation nodes calles *hosts*.

This user manual briefly describes, how to install, configure, and operate **Fitcrack**. For more detailed description of how Fitcrack works, please see our technical report [1].

- The most recent version of Fitcrack sources is available on: https://github.com/nesfit/fitcrack

II. Server installation

A. Installation tutorial

To install Fitcrack, you have to obtain all necessary prerequisities, create a user for BOINC, and configure Apache and MySQL/MariaDB database. The following tutorial shows necessary commands, which have to be run as **root**.

1) Prerequisities

- make (3.79+)
- m4 (1.4+)
- libtool (1.5+)
- autoconf (2.58+)
- automake (1.8+)
- GCC (6.3.0+)
- pkg-config (0.15+)
- Python 2 (2.2+)
- Python 3
- pip for Python 3
- MySQL (4.0.9+) or MariaDB (10.0+)
- libnotify-dev
- Apache with the following modules:
  - PHP (5+) with XML and MySQL modules
  - CGI
  - WSGI
  - rewrite
- PHP5 with cli support and the GD and MySQL modules
- OpenSSL (0.98+)

2) Installation

Create a user for running BOINC server:

```
useradd -m -c "BOINC Administrator" boincadm  -s /bin/bash
```

Create a MySQL database and user account for Fitcrack:

```
mysql -u root -p
mysql> create database fitcrack;
mysql> GRANT ALL PRIVILEGES ON fitcrack.* TO 'fitcrack'@'localhost'
       IDENTIFIED BY 'mypassword';
```

As root, use Fitcrack installer:

```
./install_fitcrack.sh
```

Or install everything manually (see README.md files for server, webadmin, and runner).

[1] https://fitcrack.fit.vutbr.cz
[2] http://www.fit.vutbr.cz/research/groups/nes@fit/.en

B. Step-by-step: Install on Debian 9 / Ubuntu 18.04 LTS

Open a **root** terminal, go to the directory with Fitcrack sources and proceed as follows:

1) Install prerequisities

Using `apt-get`, install the necessary packages:

```
apt-get install m4 make dh-autoreconf pkg-config git vim apache2 \
                libapache2-mod-php mysql-server mysql-common \
                libmysqlclient-dev zlibc zlib1g zlib1g-dev php php-xml \
                php-mysql php-cli php-gd python python python3 python-mysqldb \
                python3-pymysql python3-pip libapache2-mod-wsgi-py3 libssl-dev \
                libcurl4-openssl-dev apache2-utils libboost1.62-all-dev \
                pkg-config libnotify-dev

mysql_secure_installation # Set MySQL root password

a2enmod cgi       # enable mod CGI
a2enmod rewrite   # enable mod rewrite
a2enmod wsgi      # enable mod wsgi
systemctl restart apache2
```

2) Setup BOINC user and database

Create a UNIX user account for BOINC user, and a MySQL database and account for Fitcrack:

```
useradd -m -c "BOINC Administrator" boincadm  -s /bin/bash
mysql -u root -p
mysql> create database fitcrack;
mysql> GRANT ALL PRIVILEGES ON fitcrack.* TO 'fitcrack'@'localhost'
       IDENTIFIED BY 'mypassword';
```

3) Install Fitcrack

Run the installer:

```
./install_fitcrack.sh
```

C. Step-by-step: Install on CentOS/RHEL 7

NOTE: The following tutorial assumes SELINUX is disabled. If you wish to use SELINUX on Fitcrack server machine, you have to configure policies manually, or wait for an udpate of the tutorial.

Open a **root** terminal, go to the directory with Fitcrack sources and proceed as follows.

1) Add MariaDB 10 repository

Create file `/etc/yum.repos.d/MariaDB.repo` with the following contents:

```
[mariadb]
name = MariaDB
baseurl = http://yum.mariadb.org/10.1/centos7-amd64
gpgkey = https://yum.mariadb.org/RPM-GPG-KEY-MariaDB
gpgcheck = 1
```

2) Install prerequisities

Enable IUS, EPEL, and IUS repositories:

```
yum install -y https://$(rpm -E '%{?centos:centos}%{!?centos:rhel}%{rhel}').\
               iuscommunity.org/ius-release.rpm
yum install -y epel-release centos-release-scl
```

Install necessary packages:

```
yum install -y devtoolset-7 m4 libtool autoconf automake  git vim httpd php php-mysql \
               mod_wsgi mariadb-server mariadb-devel zlib libcurl-devel openssl-libs \
               python python36 python36u-mod_wsgi python36u-setuptools  MySQL-python \
               python2-PyMySQL boost* pkgconfig libnotify
```

Set Python 3.6 as default Python3 version:

```
alternatives --install /usr/bin/python3 python3 /usr/bin/python3.6 60
```

Install pip:

```
easy_install-3.6 pip
```

3) Configure Apache and MariaDB

Enable Apache, MariaDB, and set MariaDB root password:

```
systemctl start httpd.service
systemctl enable httpd.service
systemctl start mariadb
mysql_secure_installation # Set MariaDB root password
systemctl enable mariadb.service
```

Add BOINC user:

```
useradd -m -c "BOINC Administrator" boincadm  -s /bin/bash
```

Setup MariaDB user for Fitcrack:

```
mysql -u root -p
mysql> create database fitcrack;
mysql> GRANT ALL PRIVILES ON fitcrack.* TO 'fitcrack'@'localhost'
       IDENTIFIED BY 'mypassword';
```

4) Install Fitcrack

Switch to devtoolset 7 and run the installer:

```
scl enable devtoolset-7 bash
./install_fitcrack.sh
```

III. Server configuration and operation

A. Server daemons

To operate the server daemons, login to the server as your BOINC user, `boincadm` is default. Enter your project directory (by default `/home/boincadm/projects/fitcrack`):

```
boincadm@myserver:~$ cd projects/fitcrack
```

To get the current server **status**, type:

```
boincadm@myserver:~/projects/fitcrack$ ./bin/status
```

To **start** Fitcrack server daemons, type:

```
boincadm@myserver:~/projects/fitcrack$ ./bin/start
```

To **stop** Fitcrack server daemons, type:

```
boincadm@myserver:~/projects/fitcrack$ ./bin/stop
```

The main configuration file of Fitcrack is `config.xml`, located in your BOINC project directory. Here, it is possible to set various preferences[3] of your project. For example, you can choose, if you want to allow users to create new accounts directly from the host machines via BOINC Manager:

```
<disable_account_creation>0</disable_account_creation>
```

You can also decide, you want to set two (or more) workunits to host machines. This becomes handy in large dictionary attacks, where hosts can work on one job, and download other at the same time:

```
<max_wus_in_progress>1</max_wus_in_progress>
```

However, if you increase `max_wus_in_progress` option, you have to configure the hosts' BOINC client to compute one at a time since there is a limit on one running hahscat instance per machine. For details, see section V.

B. WebAdmin

Fitcrack **WebAdmin** is a web-based application for remote administration of the Fitcrack system. It consist of two parts: *backend*, and *frontend*, interconnected by a REST API. The WebAdmin is run by Apache HTTP server. While the front-end is a HTML+JavaScript page, the backend is a Python application using Flask, designed to be operated using WSGI.

By, default, the WebAdmin is installed automatically by Fitcrack installer. However, based on your Apache configuration, you may decide to install WebAdmin manually, or re-configure an existing installation.

The default installation locations are:

- `/var/www/html/fitcrackFE` for frontend,
- `/var/www/html/fitcrackAPI` for backend.

1) Frontend configuration

The front-end is written in VueJS, and on our github repository, we offer both sources, and pre-built version – which is used by installer, and located in `webadmin/fitcrackFE/dist` directory.

After installing the frontend, you can edit the configuration file `static/configuration.js`, and specify the IP/hostname and port, where **backend** is listening:

```
window.serverAddress = 'http://localhost:5000'
```

2) Backend configuration

The main configuration file od WebAdmin backend is `src/settings.py`. You can specify various options, e.g.

- `SQLALCHEMY_DATABASE_URI` - hostname, username, password, and database name to connect to MySQL/MariaDB server;
- `BOINC_SERVER_URI` - the URL of your BOINC project;

and others.

[3]https://boinc.berkeley.edu/trac/wiki/ProjectOptions

IV. Adding a new host

Adding a new host is pretty straightforward. All you need to do is install **BOINC Client**, and optionally **BOINC Manager**. Then, you have to create a user account on your project server, and connect to the server. If you have BOINC Manager installed, you can follow the steps in section IV-C.

A. Hardware requirements

A host machine may contain at least one OpenCL-compatible device (CPU, GPU, FPGA, DSP, co-processor...). The device should support OpenCL 1.2 runtime full profile.

B. Software requirements

Fitcrack supports hosts with 64-bit Windows, or Linux. For the device, necessary drivers have to be installed. Conretely:

1) Windows

- **NVIDIA** GPU - GeForce™ Driver,
- **AMD** GPU - AMD Radeon™ Driver,
- **Intel** CPU/GPU - OpenCL™ Runtime for Intel© Processors,

2) Linux

- **NVIDIA** GPU - GeForce™ Driver and OpenCL library. The options are:
  - Debian/Ubuntu: `nvidia-driver` and `nvidia-libopencl1` packages,
  - RHEL/CentOS/Fedora: `kmod-nvidia-*`, `nvidia-x11-drv-*` and `nvidia-libopencl1-*` packages,
  - GeForce™ Driver from NVIDIA website.
- **AMD** GPU - RadeonOpenCompute (ROCm),
- **Intel** CPU/GPU - OpenCL™ Runtime for Intel© Processors.

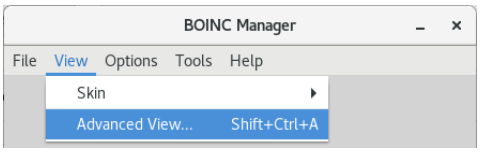
To verify if drivers were installe properly, we advise to download **hashcat**[4] binaries, and run:

```
hashcat64.bin -I
```

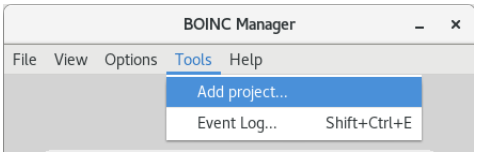
to see if all platforms and devices are detected.

C. Step-by-step guide with BOINC Manager

On your host machine, open **BOINC Manager**. We recommend switching to *advanced mode*:

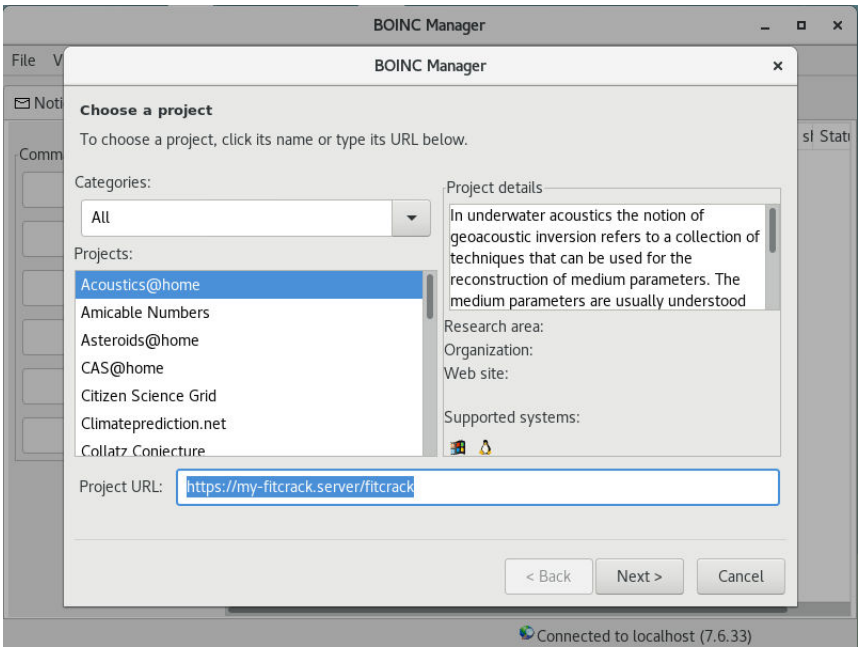


Select: **Tools** → **Add project**:

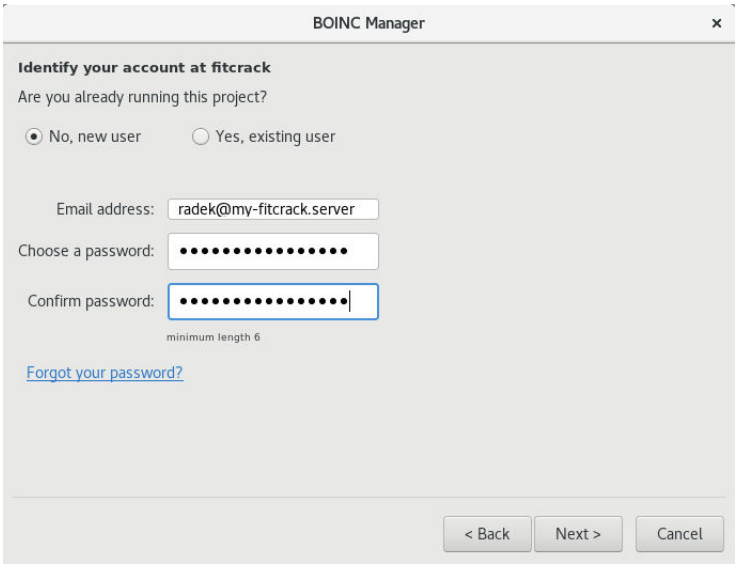


[4]https://hashcat.net/hashcat/

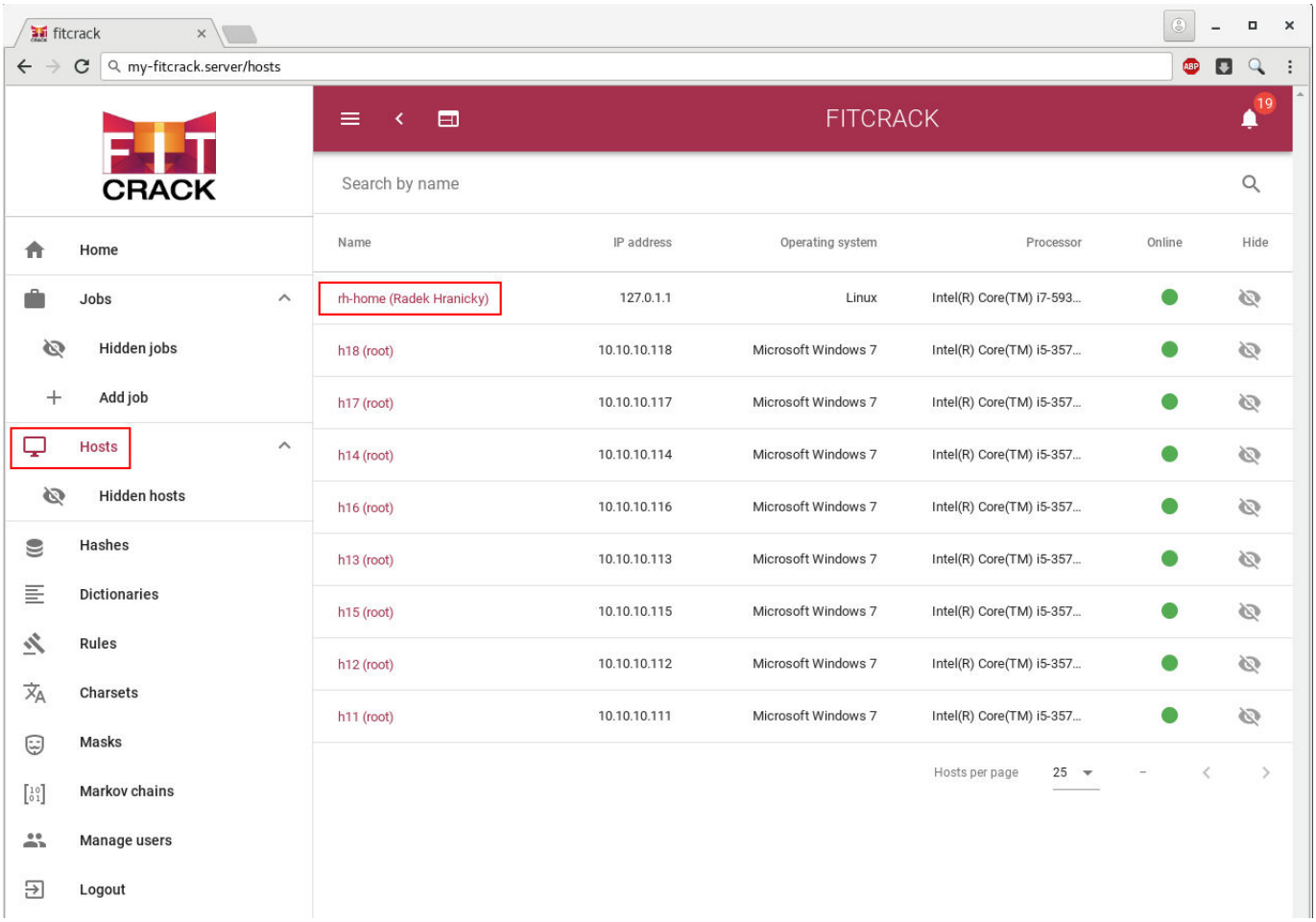
Type the **project URL**, you entered within the installation (see section II):



Create a new user account, if the feature is allowed on the server, or authenticate using an existing one:



Login to **Fitcrack WebAdmin** of your server. In **Hosts** section, you should see the newly added host:



V. Host configuration

After adding a new host (see section IV), you can optionally make a host-specific configuration file:

- `/etc/fitcrack.conf` on Linux hosts, or
- `C:\ProgramData\BOINC\fitcrack.conf` on Windows hosts.

The file will be loaded by *Runner* [1] which will pass the contents of the file as extra parameters to hashcat.

A. OpenCL device and workload configuration

By default, your host uses **all** available OpenCL devices. This may be unnecessary in some cases. E.g. you may decide to use one "weaker" GPU for display, and few "stronger" GPUs for computation.

By using **-d** parameter, you can specify the OpenCL devices (GPUs or other), which should be used. For example:

```
-d 2,3,4
```

will let hashcat use device number two, three, and four. A list of devices with corresponding numbers can be obtained by running hashcat manually with `-I` parameter, on the host machine.

You can also specify the **workload profile** for your OpenCL devices using the **-w** parameter, e.g.:

```
-w 3
```

The allowed values are between 1 (low) to 4 (nightmare). Higher values give higher cracking performance, however, at a cost of decreased latency of handling user desktop inputs, higher heat, and power consumption. For details, run hashcat with `--help` parameter.

B. Optional: multiple workunits per host

You may decide to let the server assign more than one workunit per hosts. This can dramatically reduce the overhead in large dictionary attacks, where hosts can work on one job, and download other at the same time. On server, this can be done by increasing `max_wus_in_progress` parameter in server configuration – see section III.

However, in most cases, the host machine can run only one instance od hashcat at a time. If you increase the value of workunits per progress, you **MUST** set a limit to calculate only 1 workunit at the time on **ALL HOSTS**, otherwise the extra workunits will always crash. This can be done by creating an application config file in your **BOINC Client project directory**. The directory is created automatically after connecting to a project, e.g:

- `C:\ProgramData\BOINC\projects\my-fitcrack-server.com` (Windows), or
- `/var/lib/boinc-client/projects/myserver_fitcrack` (Linux)

In BOINC Client project directory, create a file named `app_config.xml` with the following contents:

```
<app_config>
   <app>
      <name>fitcrack</name>
      <max_concurrent>1</max_concurrent>
      <report_results_immediately/>
    </app>
</app_config>
```

This will limit the host to process only one of the downloaded workunits at a time.

References

[1] Radek Hranický, Lukáš Zobal, Vojtěch Večeřa, and Matúš Múčka. *The architecture of Fitcrack distributed password cracking system*. Tech. rep. FIT-TR-2018-03, Brno, CZ, 2018, p. 30. URL: http://www.fit.vutbr.cz/research/view_pub.php.cs?id=11887.