UNIVERSITY OF TWENTE.

# SSHCure

## A Flow-Based SSH Intrusion Detection System

### User Manual

**Author** Rick Hofstede, Luuk Hendriks

**Address** University of Twente, The Netherlands

**Date** February 22, 2014

**Version** 2.3.2

# Contents

# 1 Introduction

SSHCure is a flow-based SSH intrusion detection system (IDS). As a plugin for the state-of-the-art flow collector NfSen [1], it supports and processes any NetFlow [2] and IPFIX [3]. sFlow [4] is currently not supported, as it is not a flow export technology. The core of SSHCure is an algorithm based on the work of Sperotto *et al.*, who identify and classify SSH-based dictionary attacks in three phases [5]:

1. **Scan phase** – This is usually the first phase of an SSH attack, where an attacker performs a horizontal scan over a certain IP address range.
2. **Brute-force phase** – Either immediately following the scan phase or at a later point in time, an attacker may try to login to a certain host on which it found to have a running SSH daemon. It does so by generating username/password combinations at a usually very high speed.
3. **Die-off phase** – The last phase of a dictionary attack represents the situation in which the brute-force phase has been successful for the attacker and the attacker managed to login to the target machine. From that moment the time, the target machine has been compromised and is under attacker control.

SSHCure consists of both a backend that runs the actual IDS and a frontend that aims to give operators and Computer Emergency Response Teams (CERTs) insight into the current state of their networks. This is done by means of several levels: The *Dashboard* gives an overview of attacks, top attackers/targets, etc. for a selected period of time. More details about a selected attack can be obtained from the *Attack Details* page, where a profile of the attack is shown, together with a target overview and the option to analyse the related flow data. Finally, the *Host Details* page shows in which attacks a certain host has participated, either in the role of attacker or target.

SSHCure has been optimized and tested for use in Mozilla FireFox (3+), Apple Safari (4+), Google Chrome (12+) and Microsoft Internet Explorer (7+). The SSHCure source code (and this manual) will always be made available through the SSHCure project's Web page on Sourceforge. This page is reachable by the following URLs:

- SSHCure project main page: `http://sshcure.sf.net/`
- SSHCure project download page: `http://sourceforge.net/projects/sshcure/files/`

The work on SSHCure has been supported by the following publication(s):

1. Laurens Hellemons, Luuk Hendriks, Rick Hofstede, Anna Sperotto, Ramin Sadre, Aiko Pras. *SSHCure: A Flow-Based SSH Intrusion Detection System*. In: Dependable Networks and Services. Proceedings of the 6th International Conference on Autonomous Infrastructure, Management and Security (AIMS 2012), 4-8 June 2012, Luxembourg, Luxembourg. Lecture Notes in Computer Science, Vol. 7279, ISSN 0302-9743 ISBN 978-3-642-30632-7, pp. 86-97

The following chapters cover details on the installation and configuration of SSHCure. In particular, Chapter 2 includes details on the installation that have been omitted in the readme file. Details on the configuration, combined with best practices, are included in Chapter 3. After that, Chapter 4 outlines how to use SSHCure. The last chapter of this manual, Chapter 5 explains what to do in case of any problem and discusses some frequently asked questions.

# 2 Installation

This chapter outlines details on the installation of SSHCure, which have not been included in the readme file (*readme.txt*). We refer to this file for the regular installation instructions.

## 2.1 Installation Requirements

In order to achieve the best experience when using SSHCure, the following components should be installed:

- NfSen
- Perl, together with the following modules: DBD SQLite, IP::Net, JSON, LWP
- PHP 5.2.4 or newer, together with the following modules: *mbstring*, *PDO SQLite3*

These requirements translate to the following packages:

| | **Debian/Ubuntu** | **RHEL/CentOS** | **FreeBSD** |
|---|---|---|---|
| PHP *mbstring* module | | `php-mbstring` | |
| PHP *PDO SQLite3* module | `php5-sqlite` | `php-pdo` | |
| Perl DBD SQLite module | `libdbd-sqlite3-perl` | `perl-DBD-SQLite` | `p5-DBD-SQLite` |
| Perl IP::Net module | `libnet-ip-perl` | `perl-Net-IP` | `p5-Net-IP` |
| Perl JSON module | `libjson-perl` | `perl-JSON` | `p5-JSON` |
| Perl LWP module | `libwww-perl` | `perl-libwww-perl` | `p5-libwww` |

# 3 Configuration

In most setups it will suffice to configure SSHCure in the backend, leaving the frontend configuration untouched. This chapter will therefore first describe the configuration in the backend in Chapter 3.1, followed by the frontend configuration in Chapter 3.2.

## 3.1 Backend

All backend configuration options discussed in this section can be found in the backend configuration file *config.pm*.

### 3.1.1 NfSen sources

Many NfSen-based flow collectors receive flow data from multiple flow exporters. By default, SSHCure considers the flow data from all available sources for its processing. This may, however, not yield the optimal setup and detection results for two reasons:

1. Flows may pass multiple observation points, which will result in multiple records for the same flow. This interferes with the detection algorithms of SSHCure.
2. Processing times increase as more data is used for processing.

As a best-practice, you should use only ingress/egress flow data from your edge routers/links. This can be accomplished by configuring the name of the desired data sources in the file *config.pm*. The appropriate setting name is OVERRIDE_SOURCE, which can contain one or more source names. For example, if two sources named *router1* and *router2* should be used by SSHCure, you can configure it as follows:

OVERRIDE_SOURCE = "router1:router2" (note the colon for separating the names)

SSHCure's frontend will automatically use this setting as well, when retrieving flow records from the backend or calculating statistics, for example.

### 3.1.2 Notifications

Notifications can be a simple way for informing operators/CERTs about ongoing attacks. SSHCure currently supports email and log file notifications. The configuration of notifications in *config.pm* consists of a list of *notification configs*. Every *notification config* has a name and the following parameters:

filter – Specifies for which host or set of hosts this *notification config* is specified. It can be a comma-separated list of both IP addresses and IP prefixes (and mixed). Example values: *'1.2.3.4'*, *'1.2.3.4, 5.6.7.8'* or *'1.2.3.4,5.6.7.8/16'*

filter_type – Indicates whether a notification should be sent when the host or set of hosts specified as *filter* should have the role of attacker or target. All supported values are listed in *config.pm*.

attack_phase – Indicates whether the notification should be sent when the host or set of hosts specified as *filter* enter the scan, brute-force, or compromise phase. Note that phase names are cumulative, meaning that the *notification config* will be triggered for the specified attack_phase and every attack phase that is more 'advanced'. For example, when scan attacks are specified, in fact all attacks will be considered for notification; when compromise attacks are specified, only compromise attacks are considered. All supported values are listed in *config.pm*.

when – Indicates whether the notification should be sent as soon as an attack has started (attack start), ended (attack end), or upon every change (attack update) per 5 minutes (i.e., per processing interval). As such, when notifications should be sent upon every attack update, they are sent upon attack start and attack end. All supported values are listed in *config.pm*.

notification_type – Indicates the type of notification that should be sent. Only e-mail and log files are currently supported. All supported values are listed in *config.pm*.

notification_sender – When e-mail has been specified as the *notification_type*, this field should have a single sender e-mail address. However, when log files are the preferred medium for notifications, this field should be left empty.

notification_destination – When e-mail has been specified as the *notification_type*, this field should have a comma-separated list of receiver e-mail addresses, each of them enclosed with brackets (e.g., *'<admin@domain.com>' or '<admin@domain.com>,<noc@domain.com>'*. When log files are the preferred medium for notifications, this field should have the absolute path (including file name) of the log file.

The final step in the configuration of e-mail notifications is to make sure that your server can connect to an SMTP server. If $SMTP_SERVER in *nfsen.conf* is configured as "localhost" (which is the default setting), you have to make sure that you're running an SMTP server on your machine. This can be Exim4-light with a 'smart-host' configuration, for example.

In case you have configured to write notifications to a log file, the layout of the log file is as follows:

<attack_id>,<attack_level>,<attack_start>,<attack_end>,
<attacker_ip>,<target_count>,<compromised_target_list>

Here, *<attack_id>* is the ID of the attack used in the database, *<attack_level>* can be *scan*, *brute-force* or *compromise*, *<attack_start>* and *<attack_end>* are the start and end times of the attack expressed in UNIX time, respectively, *<attacker_ip>* is the attacker's IP address in decimal notation, *<target_count>* the number of targets involved in the attack, and *<compromised_target_list>* a semi-colon-separated list of compromised targets in decimal notation (or an empty String in case no target was compromised).

Please note that SSHCure performs a sanity check of the notification configuration in *config.pm* upon every (re)start of SSHCure (triggered by a (re)start of the NfSen daemon). It is advised to check syslog for any errors or inconsistencies when (re)starting SSHCure for the first time after a configuration change.

### 3.1.3 Database maintenance

One of the key features of SSHCure is to provide an overview of hosts that participated in an attack in the role of attacker or target. To be able to do this, targets need to be stored in the database. Since the number of hosts can be very large for each attack, the size of the database will grow rapidly once SSHCure has been deployed on a high-speed link. To avoid performance problems, SSHCure supports a database routine which expires (old) database entries and performs overall database maintenance. The following settings are related to the database maintenance:

MAINTENANCE_TRIGGER – Indicates when the maintenance routine should be executed. It consists of a list of values, where each of the values consists of two parts:

1. Day of the week. Monday has index '1', Sunday has '7'. Each index consists of a single digit.
2. Hour of the day. Each index consists of two digits (including leading zero).

Both parts should be separated by a colon (:). Example: the value '("3:03", "7:03")' runs the maintenance routine each Wednesday and Sunday at 3 AM.

MAINTENANCE_TIME_NEEDED – An estimation of the duration of the maintenance in your setup in seconds. As this depends a lot on processing speed and HDD throughput of your machine, we advise to keep the default value (120).

QUICK – If quick database maintenance is enabled, the maintenance routine skips database reindexing and cleaning. As such, the maintenance will finish significantly faster. For optimal database performance, however, it is recommended to disable quick database maintenance.

In case you want to make SSHCure perform database maintenance right-away (besides the specified maintenance times), you can touch a file named *force_db_maintenance* in SSHCure's backend data directory. As soon as SSHCure detects this file during the next processing interval, the file will be removed automatically and database maintenance will be performed.

The SSHCure frontend is aware of expired targets and will indicate to the user that some information is incomplete due to database maintenance.

### 3.1.4 IP address whitelisting

SSHCure supports IP address whitelisting, which can be used in cases where benign SSH activity is reported as malicious by SSHCure. The whitelist configuration is divided into a *source* and *destination* component; IP addresses and/or IP address prefixes listed as *source*, will never be reported as an attacker, while those listed as *destination* will never be reported as target.

## 3.2 Frontend

The frontend configuration is split over two files: *config.php* and *defaults.php*. Both files are located in the */config/* directory of the SSHCure frontend. In most situations, you will not need to modify any of the frontend configuration files. However, in case you want to change the internals of the SSHCure frontend, we advise you to take the following steps:

1. Check the existing settings in *config.php*. The most common settings are listed there. If the requested setting is not listed in *config.php*, more to Step 2.
2. Check the settings in *defaults.php*. This file contains all available frontend settings. In case you want to modify any of them, please copy the corresponding line(s) to *config.php*. The settings in *defaults.php* will be overridden by the settings listed in *config.php*.

Please find here a short description on the most relevant settings.

**nfsen.config-file** – Path to your *nfsen.conf* file. The default value will fit most setups.

**backend.path** – Path to your SSHCure backend files. The default value will fit most setups.

**anonymize-ips** – If enabled, SSHCure will anonymise all IP addresses in the frontend using CryptoPAN[1]. Please note that SSHCure has not been designed to be an end-user tool and therefore IP address anonymization performed by SSHCure does not provide any SSHCure. You should only use it for the sake of demonstrations, screenshots, etc.

**ip-data-anonymized** – Enable this setting if your flow data has already been anonymized before arriving at your flow collector. It will disabled geolocation reverse DNS lookups in the frontend.

**attackprofile.maxpoints** – The Attack Details page (see Chapter 4 for more details) shows a plot of the attack profile (contacted hosts vs. time). This setting specifies the maximum number of dots in this plot. The more dots you specify, the longer it will take for your page to load. Please note that the plotting activity is completely frontend-based. Therefore, powerful machines using up-to-date Web browsers can handle more points.

**targets.maxnumber** – The Attack Details page (see Chapter 4 for more details) provides a list of attack targets. As many targets can be involved in an attack, it is wise to limit the number of targets shown in this list. Powerful machines using up-to-date Web browsers can handle longer lists. Please note that the list is sorted such that the most important attack targets are shown.

## 3.3 NfSen

Besides the required configuration steps as described in the *readme* file, there are several configuration best-practices that will boost SSHCure's performance significantly.

### 3.3.1 Dedicated profile

If SSHCure is configured to run on the 'live' profile, it considers all the data from all your sources in the SSH detection process. This is not always useful, for several reasons. For example, the dataset in the 'live' profile may be very large. To reduce the size of data to be analyzed dramatically, it is advised to use a dedicated profile for SSHCure. Such a profile can be created from the NfSen Web interface as follows:

1. Navigate to the 'Stats' tab.

[1] IP address anonymization using CryptoPAN requires the *IP::Anonymous* Perl module to be installed.

2. Enter a name for the new profile (e.g., "SSHCure").
3. Select type 'Real Profile'.
4. Select the applicable data sources.
5. Set a filter for your SSH traffic: "proto tcp and (port 22 or port 25 or port 80 or port 443 or port 6667 or port 6697)".

The next step is to change the profile reconfiguration in *nfsen.conf* so that SSHCure is solely run on the newly created profile. You can do so by opening your *nfsen.conf* file and scroll down to the @PLUGINS array. After a successful installation of SSHCure, the array should include one line related to SSHCure, such as:

[ 'live', 'SSHCure' ] (instead of 'live' you may have used '*')

Now change 'live' (or '*') into 'SSHCure' to let SSHCure solely work on your dedicated profile. After saving *nfsen.conf*, don't forget to restart the NfSen daemon to let the settings take effect.

### 3.3.2 Flow record duplicates

In case you run SSHCure on an NfSen profile with multiple sources, please make sure that flows are never exported by more than one source (i.e., make sure your NfSen profile does not contain duplicate flow records). Duplicate flow records will affect the accuracy of SSHCure's detection algorithm.

# 4 Using SSHCure

The SSHCure frontend consists of the following three pages:

**Dashboard** – The Dashboard page shows an overview over your network with respect to SSH attacks. Besides showing a list of attacks for a selected period of time, it shows statistics on top attackers and targets.

**Attack Details** – The Attack Details page provides information on one specific attack, providing general attack statistics, a list of targets and an attack profile plot.

**Host Details** – The Host Details page lists details about a specific host, either an attacker or a target. Besides geolocation and reverse DNS information, the page lists the attacks in which the host participated as an attack and/or target.

The following pages contain screenshots of each of these pages, combined with a description of their contents.

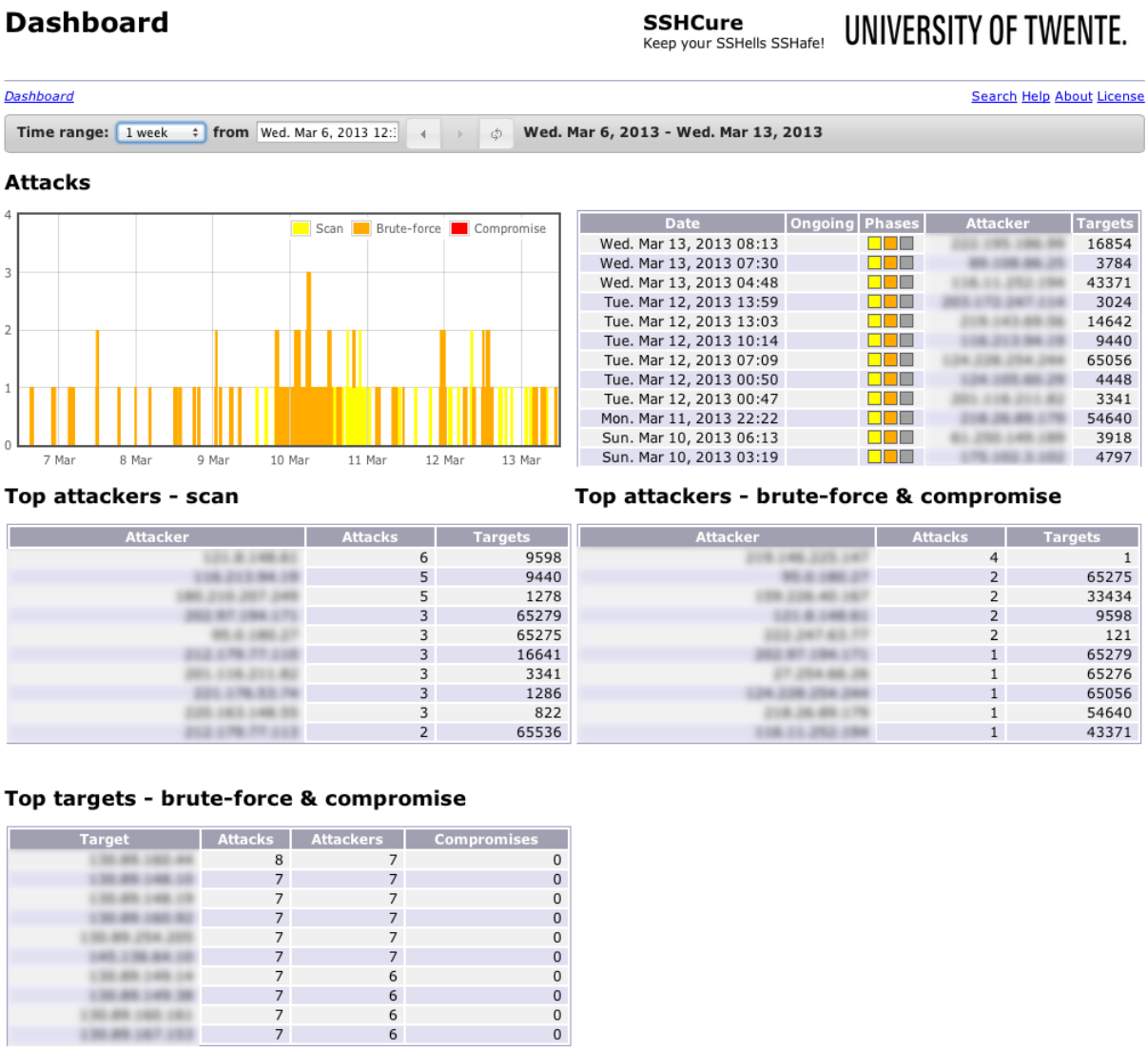


Figure 1: SSHCure's Dashboard page

At the top of the Dashboard page the time range for the page can be selected. In this screenshot, a time range of '1 week' has been selected. To the right of the time range selector a date/time selector can be found for setting the start of the time range. The following three buttons can be found next to it:

1. Back – Navigate backwards in time by the selected time range – in the case of the screenshot '1 week'.

2. Forward – Navigate forward in time by the selected time range. That button has been disabled in this screenshot, as it is not possible to navigate into the future.

3. Auto-Refresh / Now – When the currently selected time range is the latest available, the third button can be used to enabled 'auto-refresh-'. This will update your Dashboard every five minutes, as new data becomes available to NfSen. If the currently selected time range is in the past, the last button can be used to forward to 'now'.

The top left of the Dashboard page shows an attack history plot, where the number of attacks per attack type is shown for the selected time range. Next to it is a list of all attacks that shows all attacks during that occurred during the selected time range. The remaining lists shows statistics for the top scan attackers, top brute-force and compromise attackers and top targets.

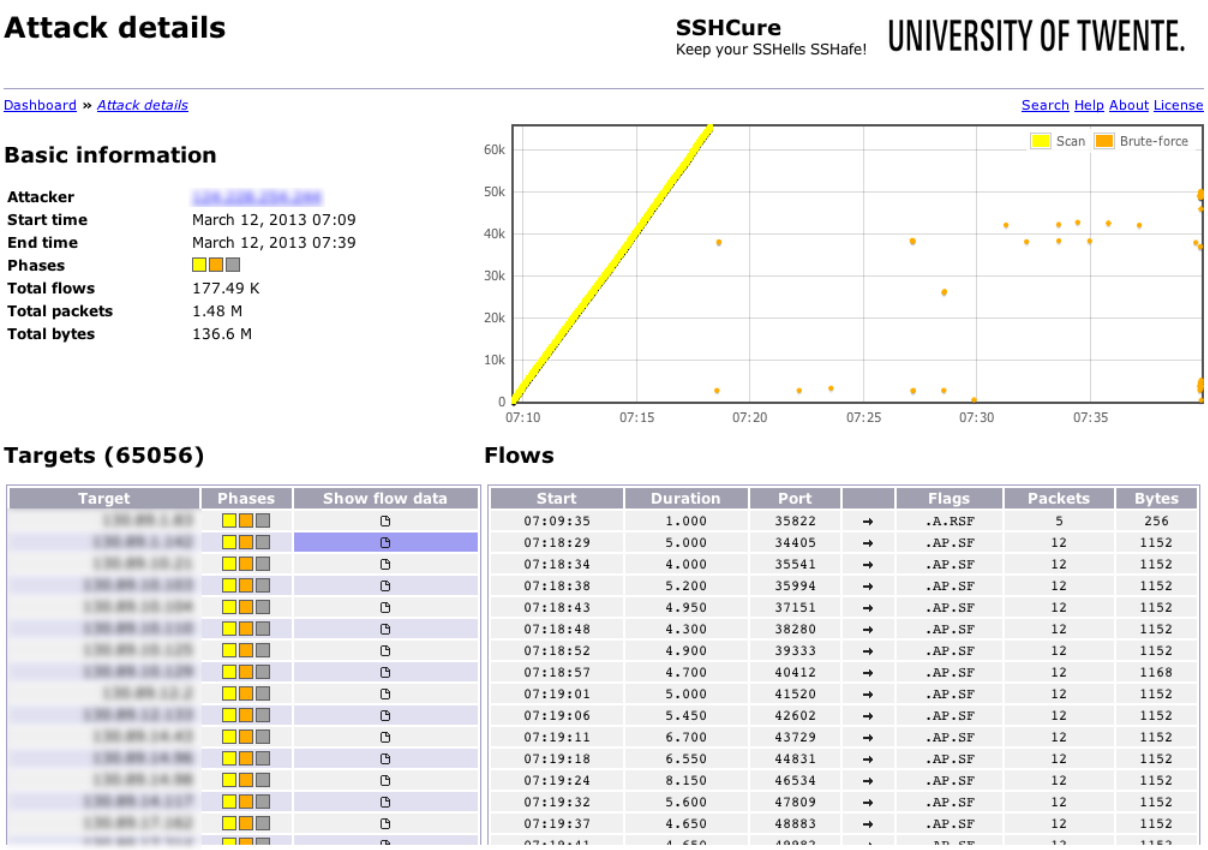


Figure 2: SSHCure's Attack Details page

The top left of the Attack Details page shows basic information on the selected attack, such as start and end time, the identified attack phases and statistics on the network traffic that was part of the attack. In the case of the screenshot, the attack consisted of both a scan and brute-force attempts, but no compromise (so the attack was not successful). To the right of the basic information is an attack profile plot, which shows the contacted hosts (y-axis) and the time (x-axis). It can be observed that a horizontal network scan took place from 07:10 – 07:18, after which several individual hosts have been brute-forced by means of a dictionary attack.

At the bottom of the page a list of targets is shown. For very large attacks, not all attacks are shown[2]. For each of the listed targets the corresponding flow data can be shown to the right of the targets listing. In the case of the screenshot, flow data is shown for the second target.

[2]See Section 3.2 for more details.

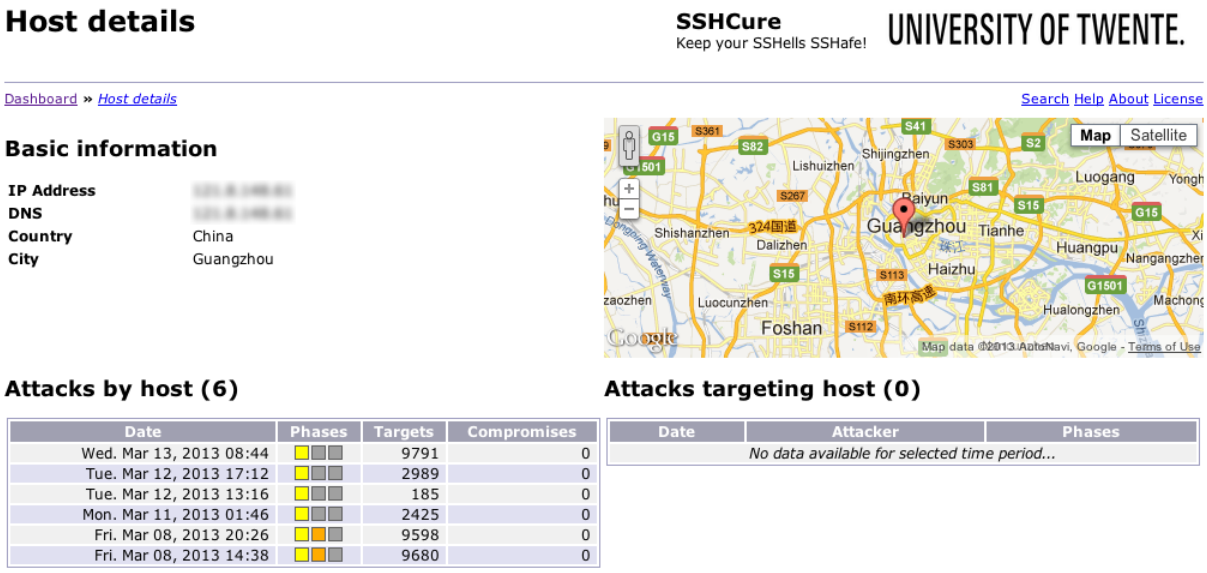


Figure 3: SSHCure's Host Details page

The third type of page within SSHCure is the Host Details page, which shows specific information on hosts (both attackers and targets). Some basic information, such as IP address, reverse DNS and geolocation information are shown at the top of the page. The bottom of the page consists of two lists: 'Attacks by host' and 'Attacks targeting hosts'. The first lists all attacks in which the selected host participated as an attacker. The second lists all attacks in which the selected host has been a target.

# 5 Troubleshooting & FAQ

If you encounter any problems with SSHCure, please perform the following steps:

1. Make sure to run SSHCure from within NfSen, instead of as a standalone application. This means that you have to run SSHCure from the *Plugins* tab in NfSen.
2. When loading SSHCure, it performs numerous checks in the backend. If any problem has been detected, a warning will be shown at the top of the Dashboard page.
3. Clear the cache of your Web browser.

Despite the fact that SSHCure has been developed with great care, you might encounter errors and/or find bugs. It could also be possible that you have ideas for improvement of SSHCure. Please help us improving SSHCure by sending an e-mail. Last but not least, we are happy to help you configuring SSHCure for use in your environment.

E-mail: `r.j.hofstede@utwente.nl`

Please do always provide as much information and details as possible when making a support request. Your support is honestly appreciated!

## 5.1 Profiling

SSHCure maintains a 'profiling database' with anonymous statistics on its performance. You can check out its contents yourselves as follows:

```
$ printf ".header on\n.mode column\n select * from profile;"
                | sqlite3 /data/nfsen/plugins/SSHCure/data/SSHCure_profile.sqlite3

time        db_size   run_time    target_count_scan  target_count_bf  target_count_do  maintenance_failed
----------  --------  ----------  -----------------  ---------------  ---------------  ------------------
1364198400  500       25          4000000            1000             3                0
```

The following parameters are stored in the 'profile' table of the database:

**time** – UNIX timestamp of the moment in which the record has been created.

**db_size** – Size of the SSHCure database (*SSHCure.sqlite3*) in bytes.

**run_time** – Processing time of the SSHCure backend (every five minutes) in seconds.

**target_count_scan** – Number of scan targets in the SSHCure database.

**target_count_bf** – Number of brute-force targets in the SSHCure database.

**target_count_do** – Number of die-off targets in the SSHCure database.

**maintenance_failed** – Indicates whether a database maintenance run has failed (1) or not (0). A reason for a run to fail can be that the processing time of the SSHCure backend is too long (see 'run_time').

**ignored_records_close_outlier** – Number of flow records within the current *nfdump* file that have starting times within a 24 hour window of the *nfdump* file timestamp, but are still clear outliers with respect to the current *nfdump* file. These flow records are ignored in the intrusion detection process.

**ignored_records_far_outlier** – Number of flow records within the current *nfdump* file that have starting times outside a 24 hour window of the *nfdump* file timestamp. These flow records are ignored in the intrusion detection process.

As the profiling database maintained by SSHCure contains solely anonymous usage statistics, we'd like to ask you to send[3] it to us periodically. This will help us to optimize SSHCure for deployment in a wider range of systems/setups.

[3] You can find the contact information at the top of this page.

## 5.2 Run lock

SSHCure uses a 'run lock' mechanism as of version 2.0. This means that if the SSHCure backend is executed while the previous execution has not yet completed, backend execution is skipped. This particular event is written to syslog as follows:

*SSHCure: Previous run has not finished yet (or a stale lock file exists); skipping data processing...*

The 'run lock' is stored in the */data/* directory of SSHCure's backend, together with the database files. In case you suspect that for whatever reason the 'run lock' has not been successfully deleted, you can remove the *run.lock* file.

## 5.3 Network traffic from SSHCure backend: Version check

As of version 2.2, SSHCure's backend performs a version check on initialization (i.e., when the NfSen daemon is started). As such, an HTTP POST message is sent out to retrieve the latest version. In case you find SSHCure generating an outbound connection, this is the reason. SSHCure is able to deal with HTTP(S) proxies, configured by means of environment variables.

## License



## Acknowledgements

This work has been supported by SURFnet's GigaPort 3 project on Next-Generation Networks and FLAMINGO, a Network of Excellence project (ICT-318488) supported by the European Commission under its Seventh Framework Programme.

# References

[1] Peter Haag, "NfSen." http://nfsen.sourceforge.net/, 2012. Accessed on 30 November 2013.

[2] B. Claise, "Cisco Systems NetFlow Services Export Version 9." RFC 3954, October 2004.

[3] G. Sadasivan, N. Brownlee, B. Claise, and J. Quittek, "Architecture for IP Flow Information Export." RFC 5470, March 2009.

[4] P. Phaal, "sFlow Version 5." http://sflow.org/sflow_version_5.txt, July 2004. Accessed on 30 November 2013.

[5] A. Sperotto, R. Sadre, P.-T. de Boer, and A. Pras, "Hidden Markov Model Modeling of SSH Brute-Force Attacks," in *20th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management (DSOM 2009)*, pp. 164–176, 2009.