Shadowserver reports automated tool

August 2016

Author:
Viktor Janevski

Supervisor(s):
Sebastian Lopienski
Stefan Lueders

CERN openlab Summer Student Report 2016

Project Specification

Every day, CERN receives mail notifications from Shadowserver, which include results of network scans for specific vulnerabilities of various types[1] for autonomous system number (ASN) 513, which is under the control of CERN.

Checking these e-mail reports manually is time-consuming and not scalable. Instead, the CERN Computer Security Team prefers some kind of a tool for:

- extracting data from e-mails (*csv.zip* attachments or embedded links to CSV files);
- confirming reports by running additional scans from inside the network;
- handling repeated reports for the same device;
- dealing with known false positives / whitelisting;
- filtering out non-CERN hosts;
- sending Security Event Management System (SEMS) notifications;
- etc.

[1] A complete list of the different report types, security vulnerabilities that Shadowserver is scanning for and more information about each of them can be found on https://www.shadowserver.org/wiki/pmwiki.php/Services/Reports

Abstract

The Shadowserver Foundation is offering a completely free-of-charge alerting and reporting service designed for ISPs, enterprises, hosting providers and other organizations that own or control a particular network space. The variety of reports provided to organizations serve as intelligence and assist in the process of locating and mitigating the security issues which occur inside their network. Being subscribed to this scanning and reporting service, CERN receives daily summaries of the security issues that happened during the past day.

Analysing and handling all the reported issues manually is a time-consuming, tedious and repetitive job, because it would require a particular person from the Computer Security Team to go through a series of steps every day. In addition, the manual approach is not scalable and tends to be error-prone, which might lead to important things being missed.

The main goal of this project is to create an automated tool that would be capable of extracting the relevant data from the received reports. However, it should not simply store the information in a database, but somehow notify the device owners that their devices were involved in a particular security issue. Also, it should be able to keep track of who was notified about what and when, in order to avoid sending multiple messages to a person about the same problem in a short period of time.

The output of the tool is a detailed report which provides an overview of the security vulnerabilities that occurred inside CERN's network during the last 24 hours, as well as a command line tool for whitelisting and managing already whitelisted devices.

Acknowledgments

Thank you to Openlab for giving me this wonderful opportunity to be part of the Openlab Summer Student Programme 2016.

Thank you to Stefan Lueders and the Computer Security Office for accepting me and treating me as an equal part of their team. I am impressed by their knowledge, skills and professionalism, and it was a real pleasure working with everyone.

Thank you to Sebastian Lopienski for providing support and constructive criticism.

Thank you to Vincent Brillaut and Liviu Valsan for their enormous patience with my endless questions.

Thank you to my parents for their endless encouragement.

Thank you to all the Summer Students for being wonderful friends.

Table of Contents

1 Introduction

1.1 Shadowserver Organization

The Shadowserver Foundation, established in 2004, is a non-profit organization comprised of world-wide professional internet security volunteers whose mission is to improve the digital security of the Internet by providing information about compromised servers, by identifying malicious attackers and by raising awareness of the spread of malicious software in general.

The activities of the foundation include gathering and reporting on malware, botnet activities and electronic fraud, or explained into more details, the organization is involved in:

- Capturing and receiving malicious software, or information related to compromised devices;
- Disassembling, sandboxing, and analysing viruses and trojans;
- Monitoring and reporting on malicious attackers;
- Tracking and reporting on botnet activities;
- Disseminating cyber threat information;
- Coordinating incident response.

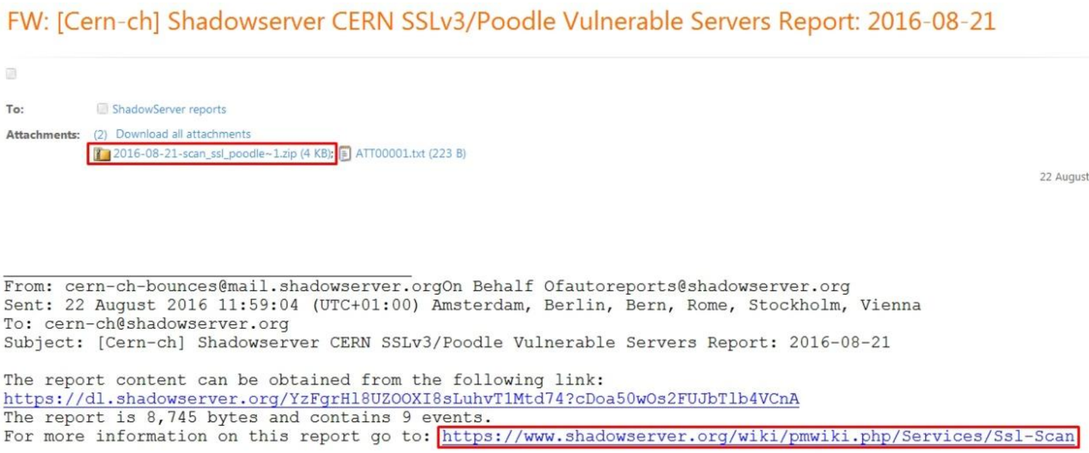
Naturally, Shadowserver does not work on its own, but collaborates with other security organizations in the process of identifying threats, but also in the phases of planning, preventing and mitigating security issues on the internet.

For the past two years, the Shadowserver Foundation has been offering a completely free-of-charge alerting and reporting service designed for ISPs, enterprises, hosting providers and other organizations that own or control a particular network space. The variety of reports provided to the outside organizations should serve as intelligence and assist in the process of locating and mitigating the security issue which had occurred inside their network. More details about Shadowserver as an organization and its work can be found on [1].

1.2 Shadowserver Periodical Reports

Being subscribed to this scanning and reporting service Shadowserver provides, CERN receives daily summaries of the computer security issues that happened during the last 24 hours. The reporting service monitors approximately a dozen different activity types and can potentially generate 45 reports, some of which are daily and others are weekly. Of course, CERN does not receive each one of them on a regular basis, simply because there is usually just a small subset of security issues that appear within its network. However, the manual analysis of these reports is not trivial and certainly does not take a negligible amount of time during the day.

Judging from the limited amount of time the Computer Security Team of CERN has been receiving these e-mail reports, the number of distinct security issue types per day is, on average, three. Each of the received e-mails contains a comma separated value (CSV) file attached to the message itself, and also a URL from where the report can be downloaded, as shown in *Figure 1*. The report files formats, or to put it more simply the attributes it contains, depend on the particular type of security issue that is being analysed. Each row represents a single security issue instance that has taken place in the previous 24 hours. A sample report sent to CERN from Shadowserver is given in *Figure 2*.



FW: [Cern-ch] Shadowserver CERN SSLv3/Poodle Vulnerable Servers Report: 2016-08-21

To: ShadowServer reports

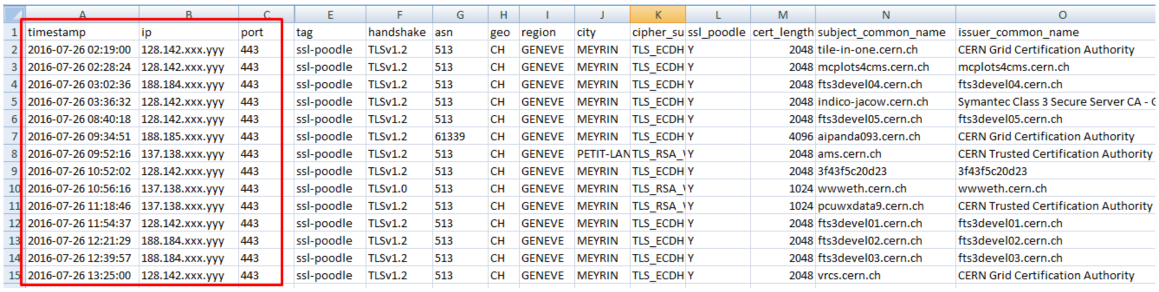
Attachments: (2) Download all attachments
2016-08-21-scan_ssl_poodle~1.zip (4 KB); ATT00001.txt (223 B)

22 August 2016 11:59

```
From: cern-ch-bounces@mail.shadowserver.orgOn Behalf Ofautoreports@shadowserver.org
Sent: 22 August 2016 11:59:04 (UTC+01:00) Amsterdam, Berlin, Bern, Rome, Stockholm, Vienna
To: cern-ch@shadowserver.org
Subject: [Cern-ch] Shadowserver CERN SSLv3/Poodle Vulnerable Servers Report: 2016-08-21

The report content can be obtained from the following link:
https://dl.shadowserver.org/YzFgrHl8UZOOXI8sLuhvT1Mtd74?cDoa50wOs2FUJbTlb4VCnA
The report is 8,745 bytes and contains 9 events.
For more information on this report go to: https://www.shadowserver.org/wiki/pmwiki.php/Services/Ssl-Scan
```

Figure 1 Sample e-mail from Shadowserver. The red rectangles show the attached report and the URL with details about the security issue type.



	A	B	C	E	F	G	H	I	J	K	L	M	N	O
1	timestamp	ip	port	tag	handshake	asn	geo	region	city	cipher_su	ssl_poodle	cert_length	subject_common_name	issuer_common_name
2	2016-07-26 02:19:00	128.142.xxx.yyy	443	ssl-poodle	TLSv1.2	513	CH	GENEVE	MEYRIN	TLS_ECDH	Y	2048	tile-in-one.cern.ch	CERN Grid Certification Authority
3	2016-07-26 02:28:24	128.142.xxx.yyy	443	ssl-poodle	TLSv1.2	513	CH	GENEVE	MEYRIN	TLS_ECDH	Y	2048	mcplots4cms.cern.ch	mcplots4cms.cern.ch
4	2016-07-26 03:02:36	188.184.xxx.yyy	443	ssl-poodle	TLSv1.2	513	CH	GENEVE	MEYRIN	TLS_ECDH	Y	2048	fts3devel04.cern.ch	fts3devel04.cern.ch
5	2016-07-26 03:36:32	128.142.xxx.yyy	443	ssl-poodle	TLSv1.2	513	CH	GENEVE	MEYRIN	TLS_ECDH	Y	2048	indico-jacow.cern.ch	Symantec Class 3 Secure Server CA - (
6	2016-07-26 08:40:18	128.142.xxx.yyy	443	ssl-poodle	TLSv1.2	513	CH	GENEVE	MEYRIN	TLS_ECDH	Y	2048	fts3devel05.cern.ch	fts3devel05.cern.ch
7	2016-07-26 09:34:51	188.185.xxx.yyy	443	ssl-poodle	TLSv1.2	61339	CH	GENEVE	MEYRIN	TLS_ECDH	Y	4096	aipanda093.cern.ch	CERN Grid Certification Authority
8	2016-07-26 09:52:16	137.138.xxx.yyy	443	ssl-poodle	TLSv1.2	513	CH	GENEVE	PETIT-LAN	TLS_RSA_\	Y	2048	ams.cern.ch	CERN Trusted Certification Authority
9	2016-07-26 10:52:02	128.142.xxx.yyy	443	ssl-poodle	TLSv1.2	513	CH	GENEVE	MEYRIN	TLS_ECDH	Y	2048	3f43f5c20d23	3f43f5c20d23
10	2016-07-26 10:56:16	137.138.xxx.yyy	443	ssl-poodle	TLSv1.0	513	CH	GENEVE	MEYRIN	TLS_RSA_\	Y	1024	wwweth.cern.ch	wwweth.cern.ch
11	2016-07-26 11:18:46	137.138.xxx.yyy	443	ssl-poodle	TLSv1.2	513	CH	GENEVE	MEYRIN	TLS_RSA_\	Y	1024	pcuwxdata9.cern.ch	CERN Trusted Certification Authority
12	2016-07-26 11:54:37	128.142.xxx.yyy	443	ssl-poodle	TLSv1.2	513	CH	GENEVE	MEYRIN	TLS_ECDH	Y	2048	fts3devel01.cern.ch	fts3devel01.cern.ch
13	2016-07-26 12:21:29	188.184.xxx.yyy	443	ssl-poodle	TLSv1.2	513	CH	GENEVE	MEYRIN	TLS_ECDH	Y	2048	fts3devel02.cern.ch	fts3devel02.cern.ch
14	2016-07-26 12:39:57	188.184.xxx.yyy	443	ssl-poodle	TLSv1.2	513	CH	GENEVE	MEYRIN	TLS_ECDH	Y	2048	fts3devel03.cern.ch	fts3devel03.cern.ch
15	2016-07-26 13:25:00	128.142.xxx.yyy	443	ssl-poodle	TLSv1.2	513	CH	GENEVE	MEYRIN	TLS_ECDH	Y	2048	vrcs.cern.ch	CERN Grid Certification Authority

Figure 2 A sample report sent to CERN by Shadowserver showing a subset of the fields that are available. The red box points out the columns whose values are used by the automated tool.

2 Shadowserver Report Analysis Tool

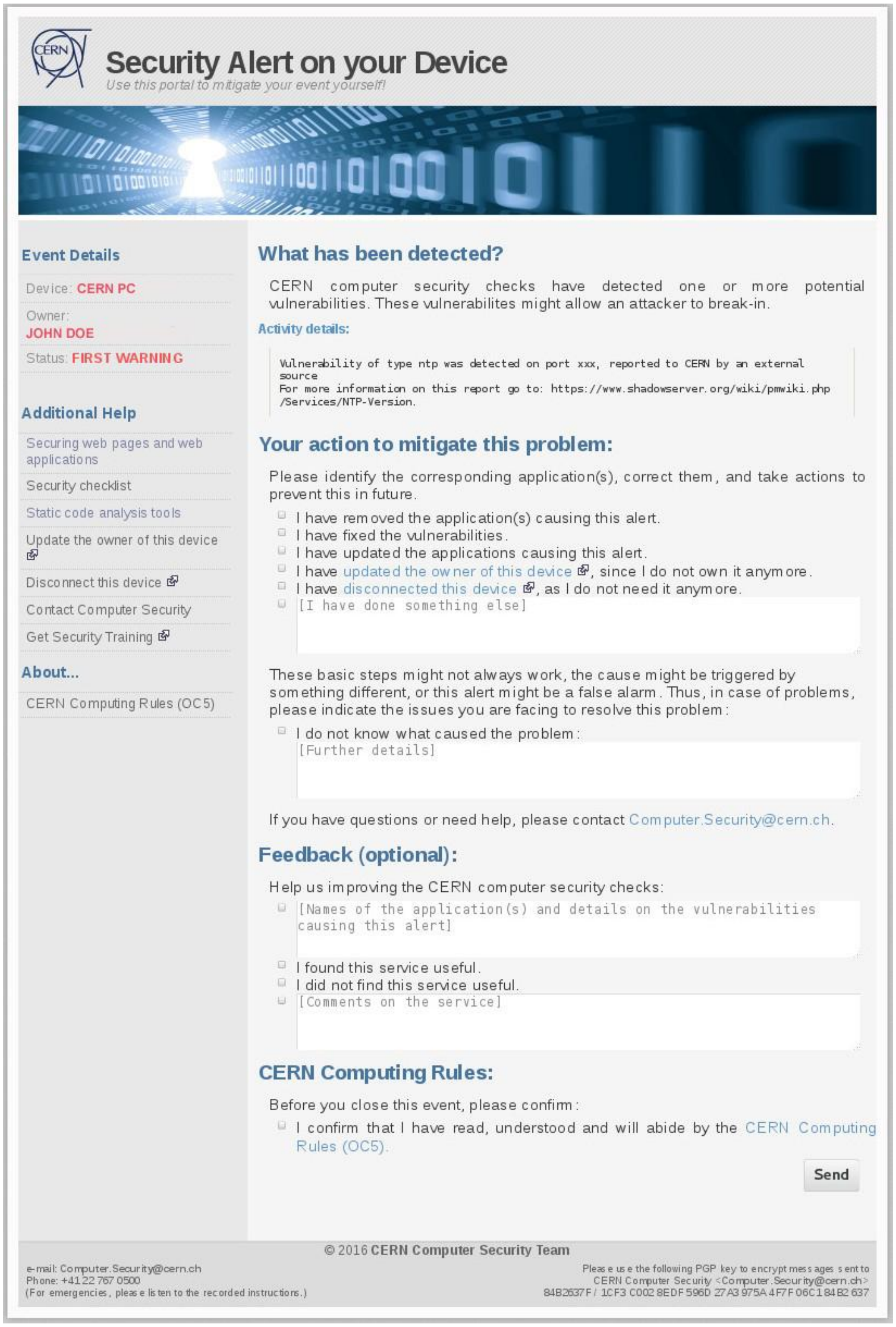
The project requirements state that the tool should be able to extract the data from the e-mails, whatever the chosen format is, by using the attachment to the message or the embedded link, find a way to handle repeated reports for the same device and deal with known false positives.

Potentially, it might be important to confirm the information provided by Shadowserver by running scans of the CERN network and perhaps filter out non-CERN hosts. In addition, being aware or informed of an existing problem is not always enough, since most of the time it is not up to the Computer Security Team to cope with the problem. Therefore, the owner of the device should be informed that they triggered a security issue which needs their attention. Furthermore, a simple notification without explaining them how to fix the issue to avoid problems in the future is not desirable. Hence, it should be agreed what the output of the tool would be — whether it will only provide a report to the Computer Security Team or should it directly inform the owners of the devices by sending them an e-mail with a status report and a possible solution or a Security Event Management System (SEMS) notification.

SEMS is a software layer between CERN's System Security DataBase (SSDB) tools and the user. SSDB is used to provide a framework to integrate all system-based security information. It is excellent for notifying the person(s) responsible for a given computing resource, in this particular case a network device. Through SSDB, by using the appropriate command, a SEMS event file can be created and published in a web folder. SEMS uses its own e-mail templates, but allows for a custom message to be sent to the device owner. A sample of a SEMS event notification page is given in *Figure 3*.

Having explained the goal of the network scanning and security issue reporting process, it is worth emphasizing that there exist a few challenges. First of all, it may sometimes take more than just 24 hours for a person to fix the security threat that has been caused. In those cases, the device's IP address will be present in multiple consecutive reports even though both the Computer Security Team and the owner of the device are well aware of the problem and the issue is being handled. Of course, sending notifications in situations like this is highly unpleasant and something that should be avoided by some kind of whitelisting, where the device owner is given a reasonable amount of time to mitigate the problem. In addition, the scanning services of Shadowserver are not without flaws, which means they could include a security issue that has already been addressed or recognized as a false positive. Ultimately, it can be agreed that a specific security issue is a false positive, that the problem it caused cannot be fixed, it is not worth the time and effort of fixing it or it is simply not a priority at the moment, so it would be a good idea to somehow remember that a particular device should not be notified about a certain security issue type, or even not notified at all in the future.

Dealing with the above mentioned issues manually would require going through each line of the CSV (comma separated value) report file for a particular issue type, comparing the reported devices against a certain file or database that stores the history of previously reported security issues, ones that are in the process of solving, as well as a whitelist, i.e. a list of devices that should not be informed in case a particular issue type appears. The necessary steps are visually better represented in *Figure 4*.



Security Alert on your Device

Use this portal to mitigate your event yourself!

Event Details

Device: **CERN PC**

Owner:
JOHN DOE

Status: **FIRST WARNING**

Additional Help

- Securing web pages and web applications
- Security checklist
- Static code analysis tools
- Update the owner of this device
- Disconnect this device
- Contact Computer Security
- Get Security Training

About...

- CERN Computing Rules (OC5)

What has been detected?

CERN computer security checks have detected one or more potential vulnerabilities. These vulnerabilites might allow an attacker to break-in.

Activity details:

```
Vulnerability of type ntp was detected on port xxx, reported to CERN by an external
source
For more information on this report go to: https://www.shadowserver.org/wiki/pmwiki.php
/Services/NTP-Version.
```

Your action to mitigate this problem:

Please identify the corresponding application(s), correct them, and take actions to prevent this in future.

□ I have removed the application(s) causing this alert.
□ I have fixed the vulnerabilities.
□ I have updated the applications causing this alert.
□ I have updated the owner of this device, since I do not own it anymore.
□ I have disconnected this device, as I do not need it anymore.
□ [I have done something else]

These basic steps might not always work, the cause might be triggered by something different, or this alert might be a false alarm. Thus, in case of problems, please indicate the issues you are facing to resolve this problem:

□ I do not know what caused the problem:
[Further details]

If you have questions or need help, please contact Computer.Security@cern.ch.

Feedback (optional):

Help us improving the CERN computer security checks:

□ [Names of the application(s) and details on the vulnerabilities causing this alert]

□ I found this service useful.
□ I did not find this service useful.
□ [Comments on the service]

CERN Computing Rules:

Before you close this event, please confirm:

□ I confirm that I have read, understood and will abide by the CERN Computing Rules (OC5).

© 2016 **CERN Computer Security Team**

e-mail: Computer.Security@cern.ch
Phone: +41 22 767 0500
(For emergencies, please listen to the recorded instructions.)

Please use the following PGP key to encrypt messages sent to
CERN Computer Security <Computer.Security@cern.ch>
84B2637F / 1CF3 C002 8EDF 596D 27A3 975A 4F7F 06C1 84B2 637

Figure 3 *A sample SEMS event page that is sent to the device owner in order to alert them about a particular security vulnerability that was detected on their device. Note, the device name and the device owner are not real.*

Clearly, the process tends to become repetitive, highly tedious and time-consuming, which would eventually lead to errors and potentially important cases being missed. In addition, the number of reports and security issues per report type might increase at some point in the future, which does not make the manual approach very scalable. Therefore, an excellent option would be to build a tool that is able to download the e-mails sent by Shadowserver, open the attached report, extract the timestamp of the security issue occurrence and any additional details of interest, and depending on the previously seen data and whitelist information, provide the Computer Security Team with a decision whether a specific host should be informed about the security issue being analysed or not, completely automatically.

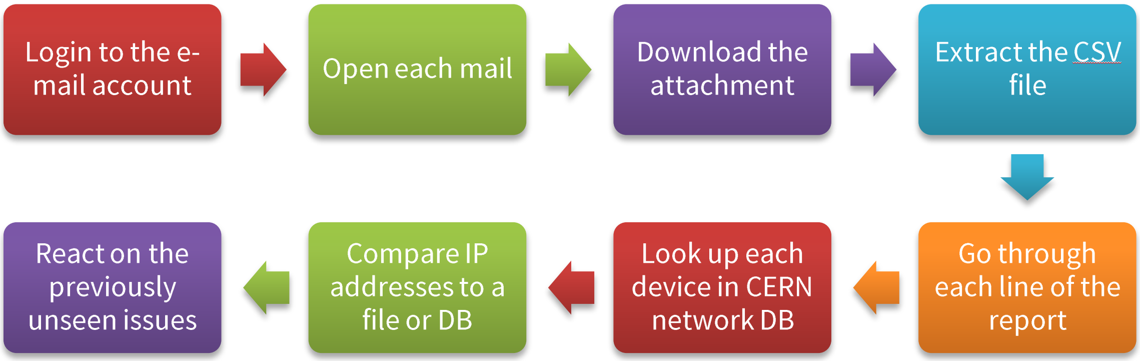

Login to the e-mail account
Open each mail
Download the attachment
Extract the CSV file
Go through each line of the report
Look up each device in CERN network DB
Compare IP addresses to a file or DB
React on the previously unseen issues

Figure 4 A diagram that shows the process of analysing Shadowserver reports. These steps that are supposed to be done manually are completely automated by the Shadowserver report analysis tool.

Also, one of the requirements of this automated tool would be to allow for whitelisting per issue type and per device, for both limited and unlimited periods of time.

3 Technical and Implementation Details

The Shadowserver e-mail report analysis tool is written entirely in Python with an underlying MySQL database used for storing the information about the previously found security issues and the whitelisted devices. The scripts are written to be executable on Python versions 2.6 and 2.7.

3.1 Database Design

The database design is made in such a way that would allow an administrator to change the time for the next notification per security vulnerability type or per IP address for a limited period of time, given in days, or forever. The database also stores information on when the security issue happened, which device caused it, when it was last reported in some Shadowserver report and the date which determines the deadline for mitigating the problem before another notification is sent. Also, the database has appropriate fields for logging when each entry was added for the first time and when its last update was, which is done automatically.

3.2 Automatic Fetching and Processing of Reports

The main entry point of the application is the *ShadowserverReports.py* script, which depending on the command line arguments given in the script call, delegates the execution to one of the two submodules: *ProcessReports.py* or *Whitelist.py*. The former is responsible for processing the reports that are sent to CERN by e-mail and the latter handles the whitelisting of devices and provides means of administering them.

The entire process of fetching the e-mails from a local folder or a remote mail server, parsing the contents of each e-mail and its attachment, as well as storing the needed data into the database is completely automated by the *ProcessReports.py* script. It does not require any settings or command line arguments, but it does depend on a configuration file where all the mail server and database connection parameters are stored. Also, this configuration file is essential in order to determine which reports should be considered recent. In other words, if device 188.185.xxx.yyy was previously reported, the configuration file should contain the number of days between the last report and the time when this device is suggested again to the administrator for attention and manual handling.

The purpose of the *MailFetch class* in the *OpenMail.py* file is to fetch the e-mails Shadowserver sends to CERN. This can be done in two ways: the e-mails could either be previously downloaded to a local folder, in an MBOX format, or they could be fetched by the script directly from a remote mail server using the IMAP4 protocol. The fetching of e-mails from a mail server is done by using *imaplib*, part of the Python standard library. The only task of the above mentioned class is to fetch the e-mails and delegate the parsing of the payload to another class, namely *MailParser* which is in the *ExtractAttachmentInformation.py* file.

The main goal of the *MailParser class* is to extract all the needed information about a specific security issue type, given a particular e-mail. The methods in this class first use regex (regular expressions) to check whether the attachment in the message has the required title format (in order to be sure that a Shadowserver report is being parsed and not something else), then the security vulnerability type name is extracted from the attachment title and the attachment file itself is extracted, unzipped and saved on the local disk, temporarily. After all new e-mail messages are processed, a method is called to iterate through all the CSV files that were extracted and saved in a local file during the previous step. In the end, the *MailParser class* reads each CSV report file, extracts the needed attribute values, calls an appropriate database method from the *DatabaseConnection class* that handles the permanent storage of data, and moves the CSV files from the temporary location to another directory that stores the entire report history.

The *DatabaseConnection class* from *DatabaseConnection.py* handles the connection to the underlying database. The class is designed in a manner that provides an interface to the other classes independently of the database used, which means that in order to migrate from a MySQL to an Oracle (or any other) database, the developer would need to make some necessary changes only inside the class itself, but that would not affect how to methods are used from external classes.

The class includes methods that can be used to establish and close a connection to the database, insert and update the table that contains information about the security issues that occurred in the past as well as the one that has the whitelisted devices. Furthermore, some of the methods in this class enable querying for particular devices in the tables and support the creation of the output report by providing the needed information.

The entire task of logging the execution steps of the programme and generating the output report about the security issues encountered in the e-mail reports sent by Shadowserver is completed by the *ReportLogging.py* script. By using the configuration files, it provides means of directing the messages to the suitable output, regardless of whether it is one of the two different log files for the tool or simply the standard output.

3.3 Device Whitelisting

The code in the *Whitelist.py* file is intended to cover the use case where Computer Security Team administrators reach a decision that a certain device should be whitelisted and not considered or notified in the future. The script itself can be executed with multiple command line arguments, depending on the problem type and whitelisting time one wants for a particular device. After the initial parsing and validation of arguments, the script calls methods which update the database.

4 Results

As mentioned in the previous sections of this document, the Shadowserver report automated tool consists of two submodules. One of them is responsible for processing the report files that are regularly sent by e-mail by Shadowserver and the other is useful for the Computer Security Team administrators, because it enables them to easily manage the whitelisted devices from the terminal.

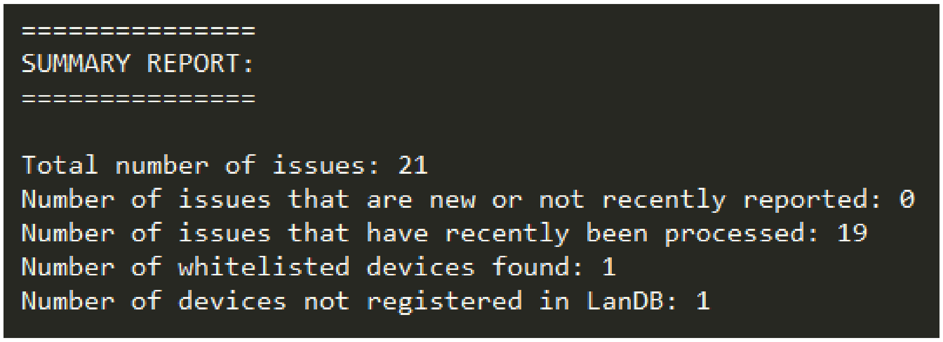
4.1 Output report

The essential result of this project is the output report, because it provides an overview of the security issues that were detected inside CERN's network during the past 24 hours. Since the scripts are intended to run completely automatically, by setting a so-called "Cron" job, the output report is sent by e-mail to the responsible members from the Computer Security Team, but also saved locally where the tool is installed, for more convenient future reference. In addition, the tool keeps logs of the execution steps and any problems that might have occurred during the running of the scripts, in order to make the processes of debugging and resolving potential issues much easier. In case the tool is run manually, the user can choose the interactive mode that is more verbose, so that all the messages and the report are also printed on the screen.

The output report contains the following sections:

- **Summary reports**: Gives information of the number of security issues that were processed by the tool, as well as the number of devices that fall into a particular case (*Figure 5*);
- **Issues to be handled manually**: List of issues given in the format *<security_issue_type>: <ip_address> - device <device_name>* that require the administrator's attention, since for the time being, the tool does not automatically create and send notifications via CERN's Security Event Management System (SEMS), but proposes a command that the administrator could run manually. *Figure 6* shows two different cases when a device is listed into this section - first when the device was reported for that particular issue a long time ago, so the device owner needs to receive a reminder and second when the device's IP address was never seen before in any Shadowserver report for the specific security vulnerability;
- **Unknown devices**: List of IP addresses that are ignored because their device names couldn't be found in the CERN network devices database, so that there is no way to contact the device owners - these devices are ignored as they are not considered to be CERN's concern. This section of the output report is shown in *Figure 7*;
- **Issues recently reported by Shadowserver**: A list of devices that were reported after the latest Shadowserver scan, but that have been previously seen and are still considered recent, according to the parameter value given in the configuration file by the tool user. This section is represented in *Figure 8*;
- **Whitelisted devices**: The devices that were found in the latest report, but have been previously added to the whitelist by an administrator, as shown in *Figure 9*;
- **Warnings and errors**: Messages that explain whether and what has gone wrong during the execution;

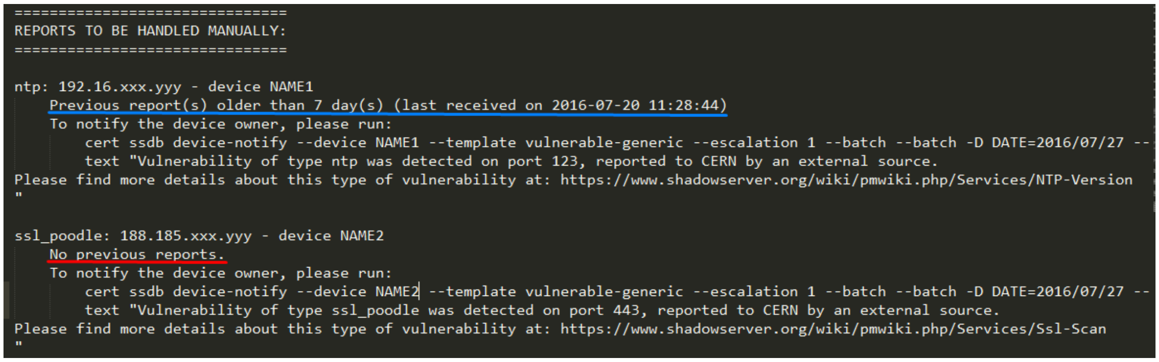
- Additional information about the report analysis process, such as: the number of new e-e-mails that were found, the attachments that were extracted and the location where the reports are saved, together with information about the number of security issues detected for each type, can be found in the log files.



```
===============
SUMMARY REPORT:
===============

Total number of issues: 21
Number of issues that are new or not recently reported: 0
Number of issues that have recently been processed: 19
Number of whitelisted devices found: 1
Number of devices not registered in LanDB: 1
```

Figure 5 Simple statistics on the security issues that were found in the analysed reports.

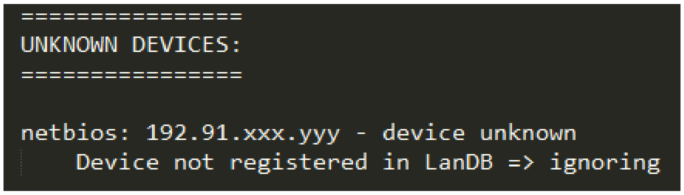


```
==============================
REPORTS TO BE HANDLED MANUALLY:
==============================

ntp: 192.16.xxx.yyy - device NAME1
    Previous report(s) older than 7 day(s) (last received on 2016-07-20 11:28:44)
    To notify the device owner, please run:
        cert ssdb device-notify --device NAME1 --template vulnerable-generic --escalation 1 --batch --batch -D DATE=2016/07/27 --
        text "Vulnerability of type ntp was detected on port 123, reported to CERN by an external source.
Please find more details about this type of vulnerability at: https://www.shadowserver.org/wiki/pmwiki.php/Services/NTP-Version
"

ssl_poodle: 188.185.xxx.yyy - device NAME2
    No previous reports.
    To notify the device owner, please run:
        cert ssdb device-notify --device NAME2 --template vulnerable-generic --escalation 1 --batch --batch -D DATE=2016/07/27 --
        text "Vulnerability of type ssl_poodle was detected on port 443, reported to CERN by an external source.
Please find more details about this type of vulnerability at: https://www.shadowserver.org/wiki/pmwiki.php/Services/Ssl-Scan
"
```

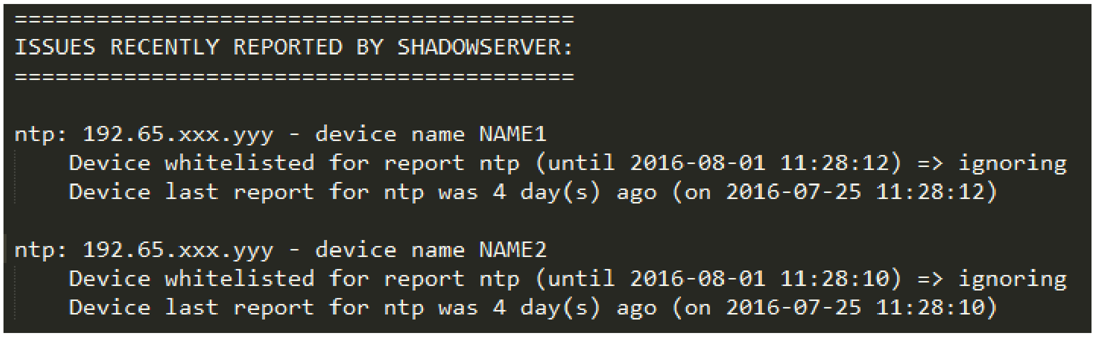
Figure 6 Issues listed in the REPORTS TO BE HANDLED MANUALLY *section require the attention of an administrator from the Computer Security Team, since the device owners need to be properly notified about the vulnerability their device has.*



```
================
UNKNOWN DEVICES:
================

netbios: 192.91.xxx.yyy - device unknown
    Device not registered in LanDB => ignoring
```

Figure 7 List of IP addresses that cannot be resolved in the network database of CERN, i.e. they are not registered devices, so there is no information about the device name, device owner or any contact.

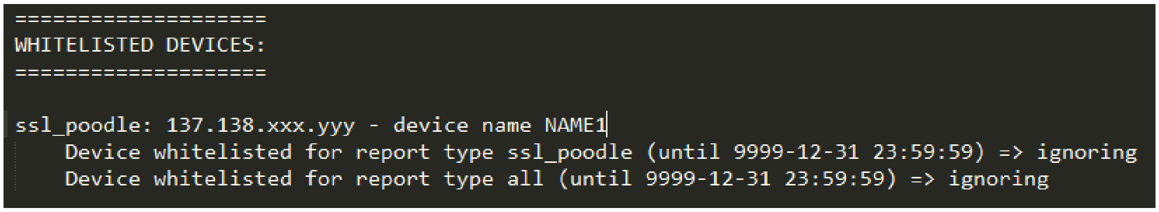


```
=========================================
ISSUES RECENTLY REPORTED BY SHADOWSERVER:
=========================================

ntp: 192.65.xxx.yyy - device name NAME1
    Device whitelisted for report ntp (until 2016-08-01 11:28:12) => ignoring
    Device last report for ntp was 4 day(s) ago (on 2016-07-25 11:28:12)

ntp: 192.65.xxx.yyy - device name NAME2
    Device whitelisted for report ntp (until 2016-08-01 11:28:10) => ignoring
    Device last report for ntp was 4 day(s) ago (on 2016-07-25 11:28:10)
```

Figure 8 Devices that were recently reported by Shadowserver, so it is not recommended to send them another notification.



```
====================
WHITELISTED DEVICES:
====================

ssl_poodle: 137.138.xxx.yyy - device name NAME1
    Device whitelisted for report type ssl_poodle (until 9999-12-31 23:59:59) => ignoring
    Device whitelisted for report type all (until 9999-12-31 23:59:59) => ignoring
```

Figure 9 Already whitelisted devices that were found in the scan report.

4.2 Whitelisting

Whenever the members of the Computer Security Team and the device owner reach a decision that the security vulnerability that has been reported by Shadowserver for that particular device cannot be solved or is simply not worth the time and effort to try and fix it, the IP address should be whitelisted so that the specific security issue is not suggested for manual handling and notifying again.

In summary, there are four possible ways a device IP address can be whitelisted:

- for a specific security issue for a limited period of time;
- for a specific security issue forever;
- for all security issues for a limited period of time;
- for all security issues forever, i.e. a notification will never be sent to this device.

In addition, the whitelisting submodule of the tool allows for complete management of the whitelist tables in a very convenient way, simply by using the terminal.

The general structure of the command to execute Whitelist.py is:

```
$ ShadowserverReports.py whitelist <action_type> [-h] [-t TYPE] [-d DAYS]
ip_address
```

Briefly, the *action_type* value can be one among four: *print, insert, modify_time* and *remove*, depending on the action that user wants to take. Then, the first argument --*type TYPE* takes a string value that can either be the name of the security issue type or "all", which would mean the specified IP address will be whitelisted for all vulnerability types. The second argument --*days DAYS* is the amount of time (number of days) the specified IP address should stay whitelisted. Passing "forever" as a value would cause the time for next notification to be set to 31.12.9999 23:59:59, which would mean the device will never receive another notification.

The following lines give an example of how the whitelisting functionality can be used.

1. Inserting a new device

```
$ ShadowserverReports.py whitelist insert -t TYPE/all -d DAYS/forever
ip_address
```

2. Modifying the next notification time

```
$ ShadowserverReports.py whitelist modify_time -t TYPE/all -d DAYS/forever
ip_address
```

3. Removing a device

```
$ ShadowserverReports.py whitelist remove -t TYPE/all ip_address
```

4. Listing certain devices

```
$ ShadowserverReports.py whitelist print -t TYPE/all -i ip_address
```

5. Listing all devices

```
$ ShadowserverReports.py whitelist print
```

5 Conclusion

The Shadowserver Foundation is continually seeking to provide timely and relevant information to the security community at large [2]. They collect a lot of different data, which does not become useful unless that information is shared. Therefore, Shadowserver is filtering that data from the network monitoring and sending it to the organizations in the form of periodical reports [3].

As a subscriber to Shadowserver's alerting and reporting service, the Computer Security Team at CERN receives e-mails that contain the results of those scans. Each document that is sent by Shadowserver contains a list of particular security vulnerabilities that were detected during the previous 24 hours inside CERN's network. Since analysing the reports manually by going through each line of every file is time-consuming and not scalable for a large network as CERN's, the Computer Security Team needs an automated tool that would run daily and provide them with a summary of the most important security events that occurred in the last day.

The Shadowserver reports automated tool is capable of fetching the new e-mails received from Shadowserver, extracting the attached report, finding all the useful and relevant information from the report, store them in a database, resolve the IP addresses to the device names, find the device owners and their contact information from the CERN network database, but also provide a suggestion whether the device owner should be notified about the security issue that their device is vulnerable to or not, depending on some previously set criterion.

The execution of the scripts results in a detailed report which contains a brief summary of what kind of issues were analysed, a list of issues that require the attention of a member of the Computer Security Team and should be manually handled, another list which gives an overview of the detected security vulnerabilities for which the tool does not recommend contacting the device owners, because it is too soon since the last notification, a section listing the devices which could not be found and resolved in CERN's network database, as well as a part where the previously whitelisted devices found in the Shadowserver report are listed. Also, for debugging purposes, the tool is keeping record of the execution steps in log files.

The previous paragraph leads to another feature of the tool which is whitelisting and managing whitelisted devices. This is useful whenever there is an agreement between the Computer Security Team and the device owner that the reported security issue cannot be mitigated, so there is no need to keep informing that person about the vulnerability.

6 Future Work

The Shadowserver reports automated tool depends heavily on the format of the e-mail messages sent by Shadowserver as well as the files attached to it, since it uses the format and naming conventions set by the organization to parse the contents and extract all the relevant information. This being said, there is a risk of having to modify the code if there is a change in any of the e-mail message body text, attachment file name, attachment format, etc. However, if this happens, the Computer Security Team will receive an alert that would notify them something has gone wrong and needs their attention.

Bibliography

[1] "Shadowserver," Shadowserver Organization, [Online]. Available: https://www.shadowserver.org/wiki/. [Accessed 25 August 2016].

[2] "Mission," Shadowserver Organization, [Online]. Available: https://www.shadowserver.org/wiki/pmwiki.php/Shadowserver/Mission. [Accessed 25 August 2016].

[3] "Reports," Shadowserver Organization, [Online]. Available: https://www.shadowserver.org/wiki/pmwiki.php/Services/Reports. [Accessed 25 August 2016].