

Detection of misconfigured HTTp services



August 1, 2016

Synchronoss technology

Galway GIS

|  |  |
| --- | --- |
| Author:  Andrej Lavrinovic | Supervasor:  Sean Boylan |

Table of Contents

[Detection of misconfigured HTTP services. 3](#_Toc459370795)

[Introduction 3](#_Toc459370796)

[Technologies 3](#_Toc459370797)

[Structure 4](#_Toc459370798)

[Performance 5](#_Toc459370799)

[Scheduling 5](#_Toc459370800)

[Input 5](#_Toc459370801)

[Nmapping 6](#_Toc459370802)

[Scanning 7](#_Toc459370803)

[Publishing 8](#_Toc459370804)

[Features 8](#_Toc459370805)

[Conclusion 9](#_Toc459370806)

# Detection of misconfigured HTTP services.

## Introduction

Configuration of web servers and web application is playing key role in the security, as these servers are responsible for serving its content and invoking applications that generate content. Misconfiguration of HTTP services may be the cause of leakage of the sensitive information, that can be used in malicious purpose later.

There wide range of IP addresses are owned or assigned to Synchronoss Technology, which are maintained by Network team. It’s important to keep all servers configured and to detect vulnerabilities before malicious people. One of the way is to request content from HTTP servers and check response for information availability.

Consider the number of hosts and the huge range of ports that may be opened it is very hard to check them manually. So ipsnapshoter tool are created to automatically scan each IP address with number of available ports. As result the html page is generated with list of snapshots for all available accesses and with all related information.

The final goal is to notify the responsible people by rising tickets if any vulnerability is detected.

## Technologies

Utility have been written in Python v.2.7 as this programming language are very powerful and suitable to reach all assigned goals. At the same time Python is interactive, with various programming approaches (such as OOP, scripting or functional), high-level language which very easy to use.

Several Python libraries have been used such as:

* subprocess (to run sub-processes)
* xml.etree (to work with xml structured file)
* SimpleHTTPServer (to publish reports)
* re (to use regular expression patterns)
* argparse (to configure execution)

and etc. for different purposes.

External projects and utilities are also used to detect vulnerabilities:

* Nmap is used to find all available ports and generate nmap xml output file.
* EyeWitness project is used for snapshots and generating of report.html page.
* Crontab is used to schedule the execution of script.

## Structure

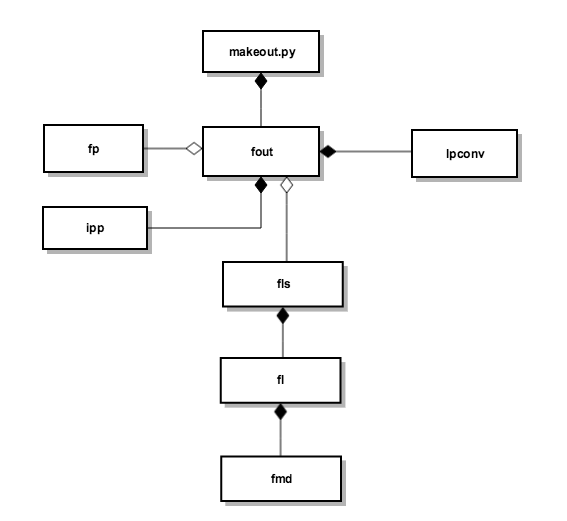


Diagram above is showing the structure of a script.

The top box ‘makeout.py’ is a main file that perform execution. Using argparse library it can accept some configuration that will affect on following process.

Process is controlled by ‘fout’ class. This class is used within makeout.py (composition). Depend of initial configuration one or another methods of fout class are called for execution.

As script is using input and output resources the appropriate paths need to be configured. ‘Fp’ (file paths) class is responsible for this. Using os.path library ‘fp’ defines its root location and builds paths to the resources, which are used in ‘fout’ (agrigation).

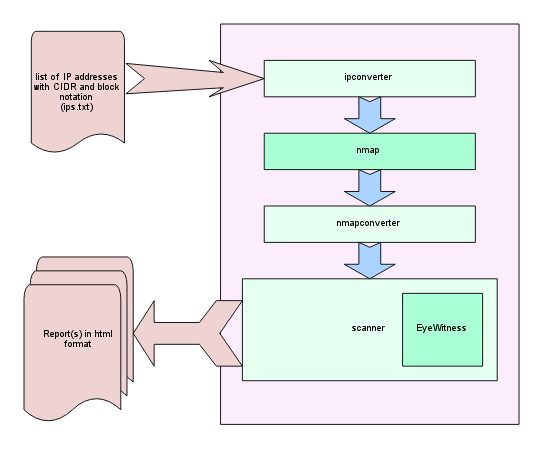
Resources represented by ‘fls’ (files) instances that accept two attributes: file name and mode (agrigation). ‘fls’ instances are used then to open file objects for its following utilisation.

‘fls’ components - file name and mode - are represented by fl class which in its turn includes ‘flmd’ class to specify mode. (composition)

During script performance some intermediated files are generating using some data components. ‘ipp’ (ip ports) class is maintaining this process. Its used by ‘fout’ (composition)

Pretty much of work is given to ‘Ipconv’ by ‘fout’ (composition). Its responsible for matching, converting, file reading and writing and more. Some algorithms are used by ‘Ipconv’.

## Performance



### Scheduling

Misconfiguration detector execution need to be performed from command line. Crontab utility is scheduling and automating performances process. Cron is a daemon that executes scheduled commands. It wakes up every minute, examining all stored crontabs. Crontab, in its turn, contain those commands and structured time point at which commands assigned to be executed.

Cron scheduler is set to be performed weekly every Sunday at 12pm.

### Input

As an input source, script is using txt file with a list of IP addresses. Different formats of IPs are accepted:

* Original IPv4 address (such as **192.168.1.0**)
* CIDR notation (such as **192.168.1.0/24**)
* Block of last 8 bits renge (such as **192.168.0-255**)
* Block of all bits range (such as **192.168.1.0 – 192.168.1.255**)

Ipconverter module of the script is responsible for understanding notation, translating all IP notations into original IPv4 and listing all IPs in initial format. If source file has heading it used by script later to specify report’s name.

It is important to know that source file must contain very last line blanked. It is necessary for detecting last IP properly as it recognized by new line character ‘\n’.

As it was mentioned above, the output is a text file with all IP addressed listed in initial format, suitable to be handled further.

### Nmapping

It is a big chance that list of IP addresses can contain thousands and thousands records. There are thousands of TCP ports that potentially may be opened. With simple maths we can calculate how many GET requests we need to make to check every possibility of vulnerability. Assume we have 1000 of IPs that need to be checked. Even if we’ll try to access top 1000 of most common TCP ports, we need to send 1000000 of GET request, and wait some while for each response. We can never end in this way especially that list may consist of much more than 1000 of IPs.

Nmap tool is used by script to solve this problem. Its determine what hosts are available on the network by sending specially crafted packets to the target host and then analyses the responses. The number of features provide the flexibility in choosing of way to scan with specifying all necessary ports.

As an output nmap returns xml file with structured information about availability of given hosts. This file is used to retrieve all hosts with open ports to further scanning.

There nmapconverter module in the script is dealing with nmap tool.

It is also responsible for:

* Preparing input instance
* Preparing all necessary state flags and other data that may be used by nmap
* Creating sub-process that runs nmap
* Scanning nmap output file
* Filtering and retrieving all useful data from nmap output file
* Converting retrieved data into list of all available IP address in initial format for further handling by next module.

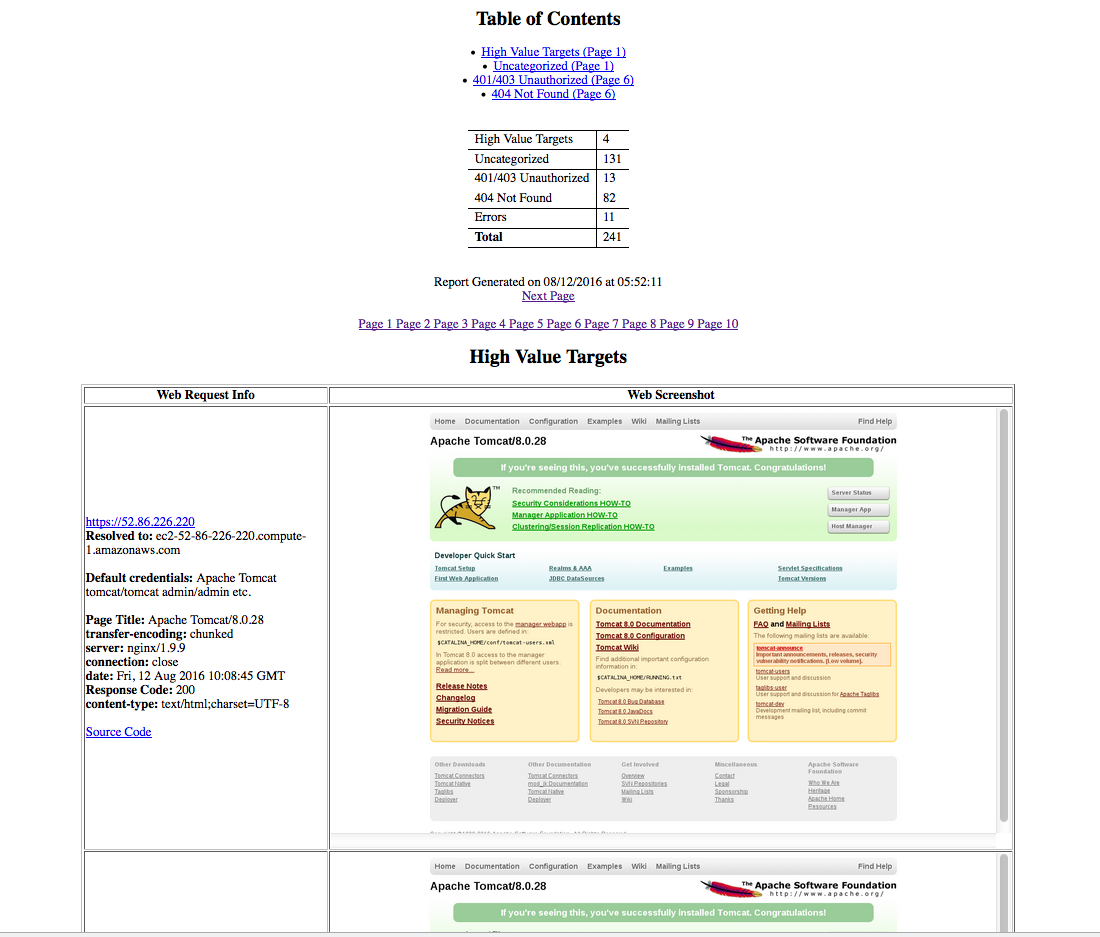
After nmapconverter module is performed we got filtered list of available hosts with available ports for access only. It is very effective way to increase performance and reduce scanning time.

### Scanning

Scan module performs the final stage in whole process. It is using another project called EyeWitness for capturing screenshots and generating reports. The tasks that are assigned to scan are:

* To prepare input instance and prepare all necessary attributes for EyeWitness execution
* To run and monitor process of scanning through all available hosts
* To brake scanning process if it to large to reduce the risk of unexpected program crash
* To manage output reports and to exclude the chance of conflict of the same report’s names

Output is a final result of performance of ipsnapshoter utility, in the form of a report in html format. It contains the relative information about HTTP server with appropriate screenshot of a GET response.



Consider the output report tester may analyse each HTTP service that is public, and can decide if there leakage or server’s misconfiguration is found. Based on decision tester made, he/she can rise a tickets for particular issue

### Publishing

To get suitable access to all reports the SimpleHTTPServer was established using appropriate python library. It creates a socket with specified port and give an access to the directory where reports are stored. Any browser can be used.

## Features

Ipsnapshoter is created to be performed in certain ways. There are few flags are configured to chose how tool need to be used:

* -p allows to specify ports to scan with. Ports need to be separated by comas with no white spaces. Port range are acceptable, for example:

./makeout.py –p 20,22,25,80-85

will run tool using default file with IPs with given ports 20, 22, 25, 80, 81, 82, 83, 84, and 85

* -f required to give a file.txt with list of IPs with no nmap scanning. For example:

./makeout.put –f /root/file/ips.txt –p 20,22,25,80-85

will run tool using given file ‘ips.txt’ with list of IPs with given ports 20, 22, 25, 80, 81, 82, 83, 84, and 85

* -n is using to make nmap scan for checking host availability. For example:

./makeout.py –n /root/file/ips.txt

will scan all available hosts with opened ports only

To reach the goals that emphasized in Introduction section is recommended to use -c flag. This configuration is using nmap scanner which scans for available hosts by exploring 2000 most commonly used TCP ports.

## Conclusion

Server Misconfiguration attacks exploit configuration weaknesses found in web and application servers. Many servers come with unnecessary default and sample files, including application files, scripts and web pages. They may also have unnecessary services enabled, such as content management, and remote administration functionality. Debugging functions may be enabled or administrative functions may be accessible to anonymous user. These features may provide a means for a hacker to bypass authentication methods and gain access to sensitive information, perhaps with elevated privileges.

Servers may include well-known default account and passwords. Failure to fully lock down or harden the server may leave improperly set file and directory permissions. Verbose and informative error messages may result in data leakage, and the information revealed could be used to formulate the next level of attack. Incorrect configurations in the server software may permit directory indexing and path traversal attacks.

The script is designed to help testers detect misconfigured HTTP servers by visually exploring received reports. Detector can handle thousands of IP addresses and response with well designed html pages that contain listed screenshots with relevant information about each available host. Its allow to security testers to find weaknesses before people with malicious intensions.

Tool was developed for specific tasks that have been assigned by Global Information Security Team in Synchronoss Technology and have been designed in particular way. However considering of architecture of the script there is possibility to improve the functionality, to add new features, to make exploitation of a detector on another platforms (Windows, X OS etc.)