Testing for Information Gathering Using OWASP Testing Guide v4 (Case Study : Udayana University SIMAK-NG Application)

Rasendriya Revo Daniswara[a1], Gusti Made Arya Sasmita[a2], I Putu Agus Eka Pratama[b3]
[a]Technology Information, Udayana University, Indonesia
[b]Technology Information, Udayana University, Indonesia
[c]Technology Information, Udayana University, Indonesia
e-mail: [1]revodaniswara@gmail.com, [2]aryasasmita@unud.ac.id, [3]eka.pratama@unud.ac.id

Abstrak

Aplikasi web adalah platform yang paling banyak digunakan untuk pengembangan sistem informasi. Peningkatan teknologi web aplikasi sebanding dengan peningkatan resikonya pula, oleh karena itu aplikasi web harus diuji terlebih dahulu untuk memastikan bahwa tidak ada resiko atau masalah keamanan pada aplikasi tersebut sebelum aplikasi tersebut dijalankan untuk public. Penetration testing adalah metode untuk menguji resiko keamanan pada aplikasi web. Langkah pertama untuk melakukan penetration testing adalah information gathering, tahap ini berguna untuk membantu penguji mengetahui spesifikasi dan vulnerability aplikasi. Penelitian ini akan melakukan implementasi dari tahap information gathering pada aplikasi SIMAK-NG Universitas Udayana menggunakan framework OWASP Testing Guide Versi 4 untuk mengetahui ada atau tidaknya permasalahan keaman pada aplikasi tersebut. Ada 10 hal yang diuji, yaitu dari OTG-INFO-001 sama OTG-INFO-010, dan hasilnya terdapat 7 pengujian bernilai positif.

Kata kunci: *penetration testing, information gathering, OWASP testing guide versi 4, keamanan siber, sistem informasi.*

Abstract

Web Application is the most used platform to develope an information system. The increased of web application technology is comparible as the risk, therefore web application must be tested first to make sure there is no risk or security issues on that application before it's launch to public. Penetration testing is a method that test the web application security risk. The first step to do penetration testing is testing for information gathering, it is used help the tester to known the specification and vulnerability of the application. This study will implement testing for information gathering to Udayana University SIMAK-NG (Academic Information System) Application using OWASP Testing Guide Version 4 framework to known there is any security issues on that application. there are ten things that were tested, that is from OTG-INFO-001 until OTG-INFO-010 and the result is seven test get positif value.

Keywords : *penetration testing, information gathering, OWASP testing guide version 4, cyber security, information system.*

1. Introduction

Web Application is the most used platform to develope an information system. However, the increased usage of websites brought up many new security issues[1], therefore web application must be tested first to make sure there is no risk or security issues on that application before it's launch to public. Security enhancements on the website can be done by testing the vulnerability of the website, one of which is the penetration testing method.[2]

Udayana university is one of many university in Indonesia that use information system technology to help organizational process on that university. Udayana university had IMISSU (Intergrated Management Information System, the Strategic of UNUD) as single sign on portal to access many application inside it. SIMAK (Information system of academic management campus) is some application inside IMISSU to help academic management system on Udayana University. SIMAK was upgraded being SIMAK-NG (SIMAK New Generation) at 2019, in April a vulnerability (that include SQL Injection, session problem, and authorization problem) was discovered. That vulnerability was reported to IT unit on Udayana University, but there is no penetration testing do until now.

The author do penetration testing on SIMAK-NG Application using OWASP Testing Guide version 4, especially on Authorization testing, Session Management Testing, and Authorization Testing. But, before that we need to do information gathering. Therefore, this study will explain the information gathering of SIMAK-NG application step.

2. Research Method / Proposed Method

The OWASP Framework used in this study is the OWASP Testing Guide framework version 4 of 2015 using the framework module: Testing for Information Gathering (OTG-INFO) where the module is dedicated to performing information gathering stages. The software/tools used is OWASP ZAP to do spidering the website and nmap to scanning network.

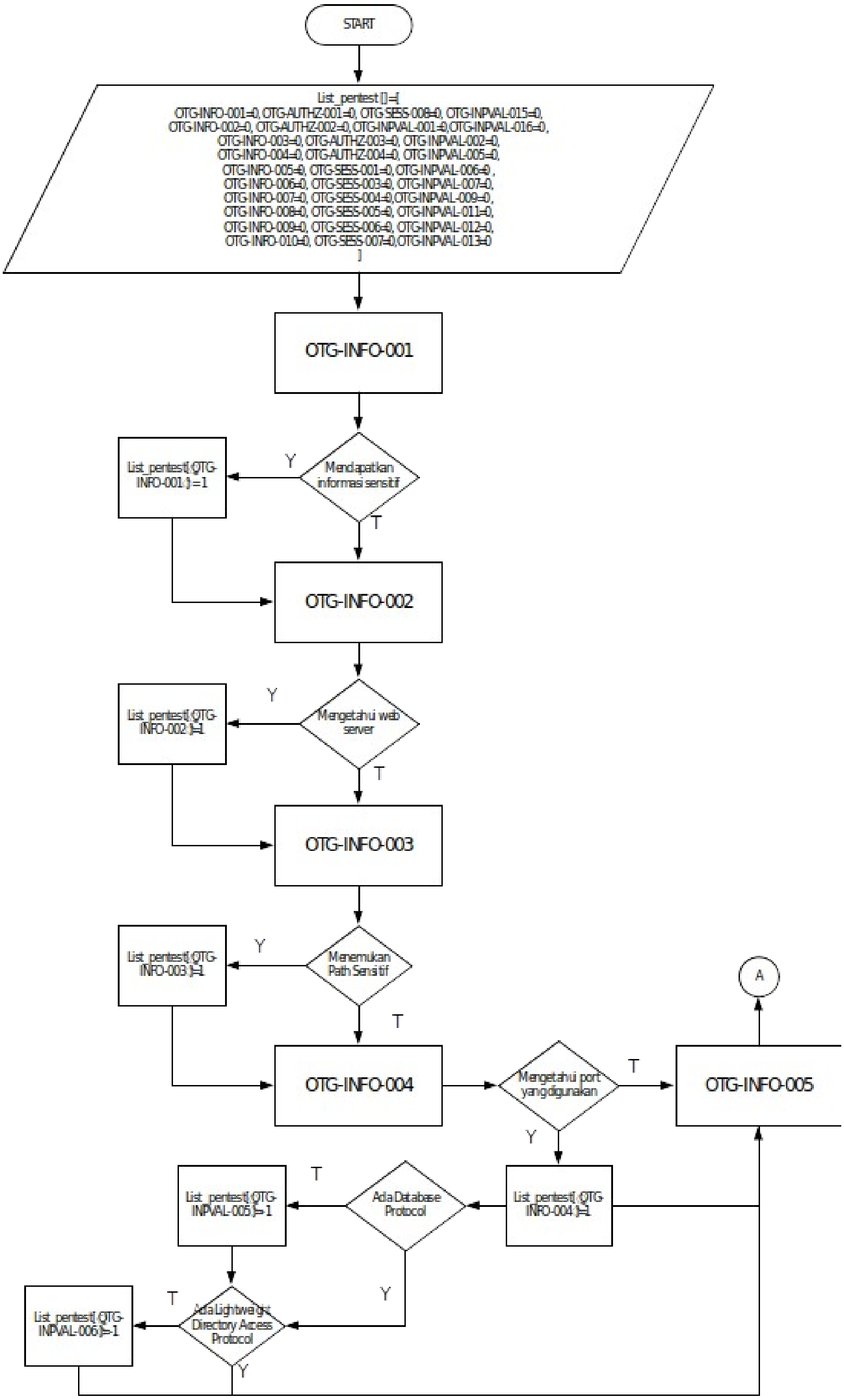

START
List_pentest []=[
OTG-INFO-001=0, OTG-AUTHZ-001=0, OTG SESS-008=0, OTG-INPVAL-015=0,
OTG-INFO-002=0, OTG-AUTHZ-002=0, OTG-INPVAL-001=0,OTG-INPVAL-016=0,
OTG-INFO-003=0, OTG-AUTHZ-003=0, OTG-INPVAL-002=0,
OTG-INFO-004=0, OTG-AUTHZ-004=0, OTG-INPVAL-005=0,
OTG-INFO-005=0, OTG-SESS-001=0, OTG-INPVAL-006=0 ,
OTG-INFO-006=0, OTG-SESS-003=0, OTG-INPVAL-007=0,
OTG-INFO-007=0, OTG-SESS-004=0,OTG-INPVAL-009=0,
OTG-INFO-008=0, OTG-SESS-005=0, OTG-INPVAL-011=0,
OTG-INFO-009=0, OTG-SESS-006=0, OTG-INPVAL-012=0,
OTG-INFO-010=0, OTG-SESS-007=0,OTG-INPVAL-013=0
]
OTG-INFO-001
Mendapatkan informasi sensitif
Y
T
List_pentest[OTG-INFO-001] =1
OTG-INFO-002
Mengetahui web server
Y
T
List_pentest[OTG-INFO-002]=1
OTG-INFO-003
Menemukan Path Sensitif
Y
T
List_pentest[OTG-INFO-003]=1
OTG-INFO-004
Mengetahui port yang digunakan
T
Y
OTG-INFO-005
A
List_pentest[OTG-INFO-004]=1
Ada Database Protocol
T
Y
List_pentest[OTG-INPVAL-005]=1
Ada Lightweight Directory Access Protocol
T
Y
List_pentest[OTG-INPVAL-006]=1

Figure 1. Information Gathering Flowchart (1).

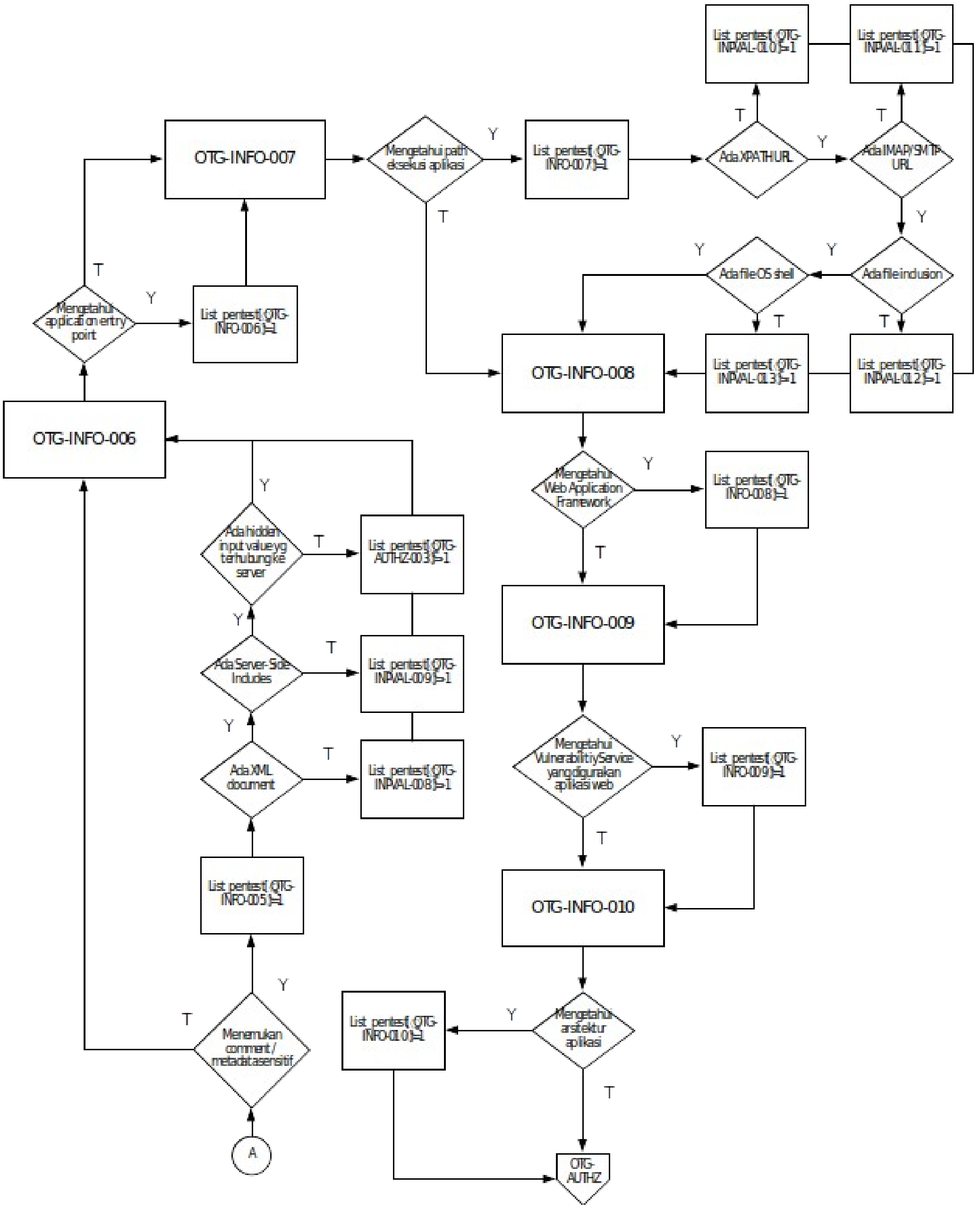

List pentest[(OTG-INPVAL-010)]=1
List pentest[(OTG-INPVAL-011)]=1
OTG-INFO-007
Mengetahui path eksekusi aplikasi
Y
List pentest[(OTG-INFO-007)]=1
T
Ada XPATH URL
Y
T
Ada IMAP/SMTP URL
Y
T
Y
Y
Ada file OS shell
Ada file inclusion
T
T
Mengetahui application entry point
Y
List pentest[(OTG-INFO-006)]=1
OTG-INFO-008
List pentest[(OTG-INPVAL-013)]=1
List pentest[(OTG-INPVAL-012)]=1
OTG-INFO-006
Mengetahui Web Application Framework
Y
List pentest[(OTG-INFO-008)]=1
Y
Ada hidden input value yg terhubung ke server
T
List pentest[(OTG-AUTHZ-003)]=1
T
OTG-INFO-009
Y
Ada Server-Side Includes
T
List pentest[(OTG-INPVAL-009)]=1
Y
Ada XML document
T
List pentest[(OTG-INPVAL-008)]=1
Mengetahui Vulnerabilitiy Service yang digunakan aplikasi web
Y
List pentest[(OTG-INFO-009)]=1
T
List pentest[(OTG-INFO-005)]=1
OTG-INFO-010
Y
T
Menemukan comment / metadata sensitif
List pentest[(OTG-INFO-010)]=1
Y
Mengetahui arsitektur aplikasi
T
A
OTG-AUTHZ

Figure 2. Information Gathering Flowchart (2).

Goals of this step is to know specification about application and to know which step that cant be done. The result of this testing represented as -1,0, or 1 state. -1 state means the test cant be done cause there is some requirement to test is not found, 0 state means the test is unsuccess, and 1 state means the test is successfully.

The default value of result of testing is 0, then the testing result will change the value. The testing start from OTG-INFO-001 until OTG-INFO-010 like methodology flowchart in figure 1, if the resulting is “positive” the value will changes to 1. Sometime there is some testing that relate to another testing, if the first testing is “negative” the value is still 0, and it means the next testing is can’t perform, the value for the next testing being -1. For example if the result from OTG-INFO-004 is not found any database port, it means OTG-INPVAL-005 can’t be perform.

The final result of information gathering testing represented as -1,0, and 1. If the result is 1that will explain what is the risk from that information.

3. Literature Study

3.1. Penetration Testing

The penetration testing method or often called “pentest” is the practice of computer system, network, or web application security testing to find security vulnerabilities that can be exploited by attackers by providing stages of system attacks to the system.[3]

3.2. Black Box Pentesting

This form testing is the opposite of White Box. An attacker who is performing a Black Box pen test needs to have stealth and not have their cover blown.[4]

3.3. OWASP Testing Guide Version 4

OWASP is a non-profit organization that focuses on improving software security [5]. OWASP provides many tools, guides and testing methodologies for cyber security under an open source license, specifically the OWASP Testing Guide (OTG). [6] The OTG is divided into three main parts including the OWASP testing framework for web application development, web application testing methodology, and system evaluation reporting. The web application testing methodology can be used independently, or can be used as a testing framework. A web application developer can use the framework to build web applications by considering the protection and security aspects followed by security testing with the penetration testing method to test the system security of the web application developed.[7]

3.4. OWASP ZAP

Zed Attack Proxy (ZAP) from OWASP is one of the most popular free security scanning tools in the world and is actively managed by hundreds of international volunteers. ZAP can automatically scan for security vulnerabilities in web applications when they are developed and tested. ZAP is a reliable tool for experienced penetration testers to be used as automatic safety testing tools [8][9]

3.5. Nmap

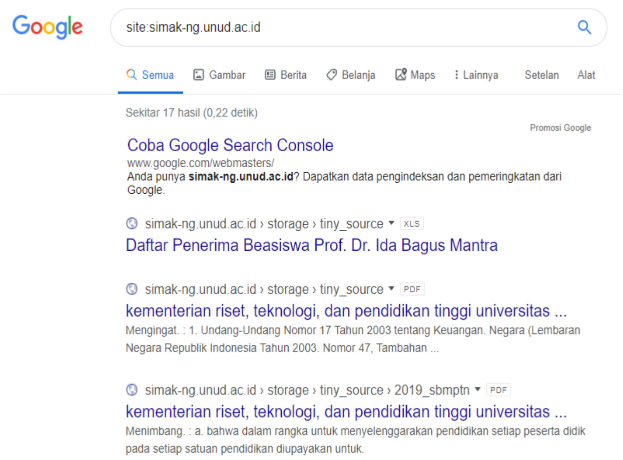
Nmap ("Network Mapper") is a free and open source (license) utility for network discovery and security auditing. Many systems and network administrators also find it useful for tasks such as network inventory, managing service upgrade schedules, and monitoring host or service uptime. Nmap uses raw IP packets in novel ways to determine what hosts are available on the network, what services (application name and version) those hosts are offering, what operating systems (and OS versions) they are running, what type of packet filters/firewalls are in use, and dozens of other characteristics. It was designed to rapidly scan large networks, but works fine against single hosts. Nmap runs on all major computer operating systems, and official binary packages are available for Linux, Windows, and Mac OS X.[10]

4. Result and Discussion

This section will describe a process and result for each testing for information gathering on this research.

4.1. Conduct Search engine discovery/ reconnaissance for information leakage (OTG-INFO-001)

This testing use google search engine to test there is vulnerability / sensitive information found or not. To help this testing we use google search operator.



site:simak-ng.unud.ac.id

Sekitar 17 hasil (0,22 detik)

Coba Google Search Console
www.google.com/webmasters/
Anda punya **simak-ng.unud.ac.id**? Dapatkan data pengindeksan dan pemeringkatan dari Google.

simak-ng.unud.ac.id › storage › tiny_source ▾ XLS
Daftar Penerima Beasiswa Prof. Dr. Ida Bagus Mantra

simak-ng.unud.ac.id › storage › tiny_source ▾ PDF
kementerian riset, teknologi, dan pendidikan tinggi universitas ...
Mengingat. : 1. Undang-Undang Nomor 17 Tahun 2003 tentang Keuangan. Negara (Lembaran Negara Republik Indonesia Tahun 2003. Nomor 47, Tambahan ...

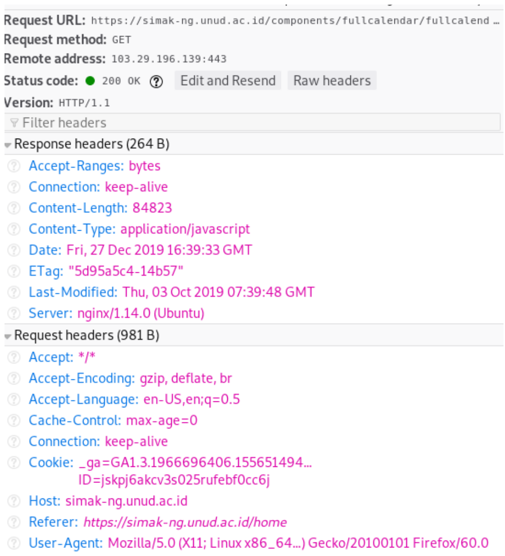
simak-ng.unud.ac.id › storage › tiny_source › 2019_sbmptn ▾ PDF
kementerian riset, teknologi, dan pendidikan tinggi universitas ...
Menimbang. : a. bahwa dalam rangka untuk menyelenggarakan pendidikan setiap peserta didik pada setiap satuan pendidikan diupayakan untuk.

Figure 3. Google Search Operator.

Figure 3 is the search result from google search engine using keyword "site:simak-ng.unud.ac.id" to know all result relate about that application. The result is only 17 url relate about SIMAK-NG application and it's only PDF and XLS document that is not contain any sensitive information / vulnerability.

4.2. Fingerprint Web Server (OTG-INFO-002)

This section use inspect element from browser to see response header application. Figure 4 is the response header SIMAK-NG Application, from that we know this application using nginx/1.14.0/ubuntu as web server.

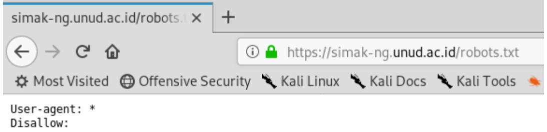


```
Request URL: https://simak-ng.unud.ac.id/components/fullcalendar/fullcalend …
Request method: GET
Remote address: 103.29.196.139:443
Status code: ● 200 OK
Version: HTTP/1.1
Response headers (264 B)
  Accept-Ranges: bytes
  Connection: keep-alive
  Content-Length: 84823
  Content-Type: application/javascript
  Date: Fri, 27 Dec 2019 16:39:33 GMT
  ETag: "5d95a5c4-14b57"
  Last-Modified: Thu, 03 Oct 2019 07:39:48 GMT
  Server: nginx/1.14.0 (Ubuntu)
Request headers (981 B)
  Accept: */*
  Accept-Encoding: gzip, deflate, br
  Accept-Language: en-US,en;q=0.5
  Cache-Control: max-age=0
  Connection: keep-alive
  Cookie: _ga=GA1.3.1966696406.155651494…
          ID=jskpj6akcv3s025rufebf0cc6j
  Host: simak-ng.unud.ac.id
  Referer: https://simak-ng.unud.ac.id/home
  User-Agent: Mozilla/5.0 (X11; Linux x86_64…) Gecko/20100101 Firefox/60.0
```

Figure 4. Fingerprint web server.

4.3. Review Webserver Metafiles for Information Leakage (OTG-INFO-003)

This test try to find robots.txt file on SIMAK-NG application and see there is any sensitive url or not. This file will found on https://simak-ng.unud.ac.id/robots.txt.


simak-ng.unud.ac.id/robots.
https://simak-ng.unud.ac.id/robots.txt
Most Visited
Offensive Security
Kali Linux
Kali Docs
Kali Tools

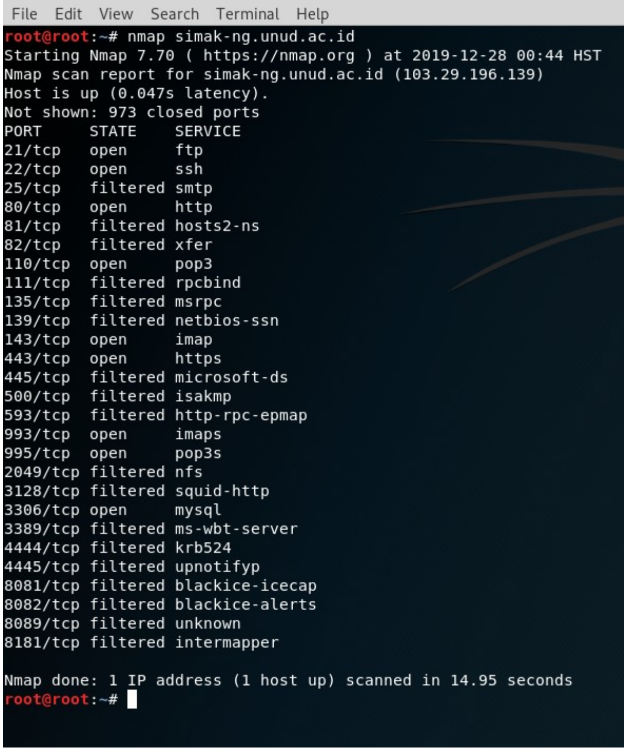
```
User-agent: *
Disallow:
```

Figure 5. Robots.txt.

Figure 5 is the robots.txt file on SIMAK-NG Application and we know there's nothing found.

4.4. Enumerate Application on Webserver (OTG-INFO-004)

Enumerate Application testing use nmap application to scanning SIMAK-NG Application port.



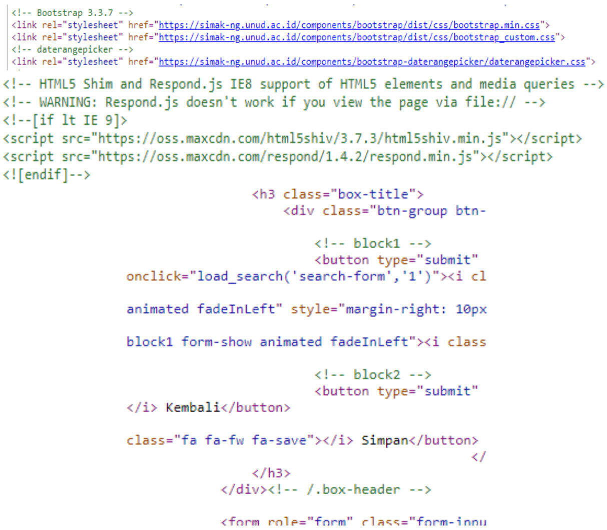
```
root@root:~# nmap simak-ng.unud.ac.id
Starting Nmap 7.70 ( https://nmap.org ) at 2019-12-28 00:44 HST
Nmap scan report for simak-ng.unud.ac.id (103.29.196.139)
Host is up (0.047s latency).
Not shown: 973 closed ports
PORT      STATE    SERVICE
21/tcp    open     ftp
22/tcp    open     ssh
25/tcp    filtered smtp
80/tcp    open     http
81/tcp    filtered hosts2-ns
82/tcp    filtered xfer
110/tcp   open     pop3
111/tcp   filtered rpcbind
135/tcp   filtered msrpc
139/tcp   filtered netbios-ssn
143/tcp   open     imap
443/tcp   open     https
445/tcp   filtered microsoft-ds
500/tcp   filtered isakmp
593/tcp   filtered http-rpc-epmap
993/tcp   open     imaps
995/tcp   open     pop3s
2049/tcp  filtered nfs
3128/tcp  filtered squid-http
3306/tcp  open     mysql
3389/tcp  filtered ms-wbt-server
4444/tcp  filtered krb524
4445/tcp  filtered upnotifyp
8081/tcp  filtered blackice-icecap
8082/tcp  filtered blackice-alerts
8089/tcp  filtered unknown
8181/tcp  filtered intermapper

Nmap done: 1 IP address (1 host up) scanned in 14.95 seconds
root@root:~#
```

Figure 6. Enumerate.

Figure 6 is the result of port scanning using nmap, we know there is 9 open port and 18 filtered port.

4.5. Review webpage comments and metadata for information leakage (OTG-INFO-005)

This step will search hidden input / HTML comment from HTML source that is contain any sensitive information.



```
<!-- Bootstrap 3.3.7 -->
<link rel="stylesheet" href="https://simak-ng.unud.ac.id/components/bootstrap/dist/css/bootstrap.min.css">
<link rel="stylesheet" href="https://simak-ng.unud.ac.id/components/bootstrap/dist/css/bootstrap_custom.css">
<!-- daterangepicker -->
<link rel="stylesheet" href="https://simak-ng.unud.ac.id/components/bootstrap-daterangepicker/daterangepicker.css">
<!-- HTML5 Shim and Respond.js IE8 support of HTML5 elements and media queries -->
<!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
<!--[if lt IE 9]>
<script src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>
<script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
<![endif]-->
                                <h3 class="box-title">
                                    <div class="btn-group btn-

                                        <!-- block1 -->
                                        <button type="submit"
                onclick="load_search('search-form','1')"><i cl

                animated fadeInLeft" style="margin-right: 10px

                block1 form-show animated fadeInLeft"><i class

                                        <!-- block2 -->
                                        <button type="submit"
                </i> Kembali</button>

                class="fa fa-fw fa-save"></i> Simpan</button>
                                                              </
                                </h3>
                            </div><!-- /.box-header -->

                            <form role="form" class="form-inpu
```

Figure 6. HTML Comment.

There is no information sensitive found, hidden input not found and the HTML comment is only contain information about start / end html tag.

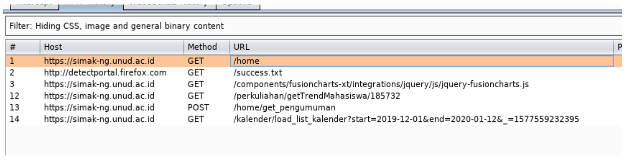
4.6. Identify application entry points (OTG-INFO-006)

There is many entry point on SIMAK-NG application that's maybe can used as attack vector. "Profil Mahasiswa" page contain information about student that can be edited, of course there is many input on that page will send to server.

There is many page that used to search something like list of study, list of lecture, etc. that search input maybe can be use as attack vector. Other than there is many page that include feature to upload document, for example to upload final project, report project, etc. that input to upload can be use as attack vector too.

4.7. Map execution paths through application (OTG-INFO-007)

Knowing execution paths of SIMAK-NG application can be known by explore any page using burpsuite.



Filter: Hiding CSS, image and general binary content

#	Host	Method	URL	P
1	https://simak-ng.unud.ac.id	GET	/home	
2	http://detectportal.firefox.com	GET	/success.txt	
3	https://simak-ng.unud.ac.id	GET	/components/fusioncharts-xt/integrations/jquery/js/jquery-fusioncharts.js	
12	https://simak-ng.unud.ac.id	GET	/perkuliahan/getTrendMahasiswa/185732	
13	https://simak-ng.unud.ac.id	POST	/home/get_pengumuman	
14	https://simak-ng.unud.ac.id	GET	/kalender/load_list_kalender?start=2019-12-01&end=2020-01-12&_=1577559232395	

Figure 7. Mapping using execution path using burpsuite.

Figure 7 is sample of using burpsuite to know map execution path on SIMAK-NG application.

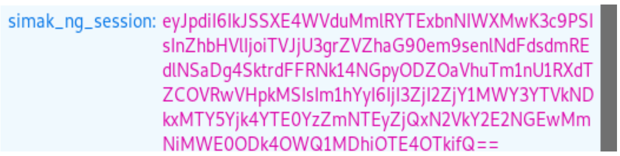
Table 1. List of Path Execution.

URL	Description
http://simak-ng.unud.ac.id/home	The main of home page.
http://simak-ng.unud.ac.id/perkuliahan/getTrendMahasiswa/[parameter]	Get student trend data.
http://simak-ng.unud.ac.id/home/get_pengumuman	Get announcement data.
http://simak-ng.unud.ac.id/mahasiswa/profile	The main of profile page.
http://simak-ng.unud.ac.id/getProfile/[token]	Get student profile data.
http://simak-ng.unud.ac.id/mahasiswa/matakuliah	The main of study page.
http://simak-ng.unud.ac.id/mahasiswa/getMatakuliah?[Parameter]	Get study data.
http://simak-ng.unud.ac.id/mahasiswa/matakuliahTawar	The main of "fixed study" page.
http://simak-ng.unud.ac.id/mahasiswa/getMatakuliahTawar?[Parameter]	Get "fixed study" data.
http://simak-ng.unud.ac.id/KerjaPraktek/KpPengajuan	The main of request internship page.

Table 1 is the sample list of map path execution on SIMAK-NG Application. there is all get method tracked from burpsuite.

4.8. Fingerprint Web Application Framework (OTG-INFO-008)

If we try to see the session cookies from browser it will like figure 9.



```
simak_ng_session: eyJpdiI6IkJSSXE4WVduMmlRYTExbnNIWXMwK3c9PSI
                  sInZhbHVlIjoiTVJjU3grZVZhaG90em9senlNdFdsdmRE
                  dlNSaDg4SktrdFFRNk14NGpyODZOaVhuTm1nU1RXdT
                  ZCOVRwVHpkMSIsIm1hYyI6IjI3ZjI2ZjY1MWY3YTVkND
                  kxMTY5Yjk4YTE0YzZmNTEyZjQxN2VkY2E2NGEwMm
                  NiMWE0ODk4OWQ1MDhiOTE4OTkifQ==
```

Figure9. Session Cookies.

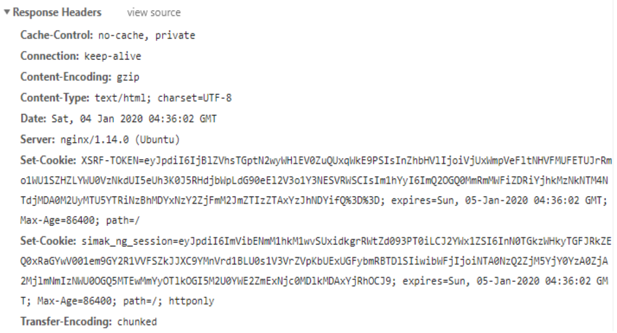
There is base 64 encryption, the decryption of that's string is will be like that :

```
{"iv":"BRIq8YWn2iQa11nsHYs0+w==",
"value":"MRcSx+eVahotzolzyMtWlvdDv
SRh88JKktQQ6Mx4jr86NiXnNmgSTWu
6B9TpTzd1",
"mac":"27f26f651f7a5d491169b98a14c6f
512f417edca64a02cb1a48989d508b91899"}
```

From that we know there is AES-CBC encryption that is like Laravel encryption method. So we can make assumption SIMAK-NG Application using Laravel framework.

4.9. Fingerprint Web Application (OTG-INFO-009)

HTTP header contain any information about web server like figure 10. From HTTP Header we know the web server used is nginx/1.14.0 (ubuntu). Other than that we can review HTML Source to know there is any web service with known vulnerability or not.



```
▾ Response Headers     view source
  Cache-Control: no-cache, private
  Connection: keep-alive
  Content-Encoding: gzip
  Content-Type: text/html; charset=UTF-8
  Date: Sat, 04 Jan 2020 04:36:02 GMT
  Server: nginx/1.14.0 (Ubuntu)
  Set-Cookie: XSRF-TOKEN=eyJpdiI6IjBlZVhsTGptN2wyWHlEV0ZuQUxqWkE9PSIsInZhbHVlIjoiVjUxWmpVeFltNHVFMUFETUJrRm
  o1WU1SZHZLYWU0VzNkdUI5eUh3K0J5RHdjbWpLdG90eEl2V3o1Y3NESVRWSCIsIm1hYyI6ImQ2OGQ0MmRmMWFiZDRiYjhkMzNkNTM4N
  TdjMDA0M2UyMTU5YTRiNzBhMDYxNzY2ZjFmM2JmZTIzZTAxYzJhNDYifQ%3D%3D; expires=Sun, 05-Jan-2020 04:36:02 GMT;
  Max-Age=86400; path=/
  Set-Cookie: simak_ng_session=eyJpdiI6ImVibENmM1hkM1wvSUxidkgrRWtZd093PT0iLCJ2YWx1ZSI6InN0TGkzWHkyTGFJRkZE
  Q0xRaGYwV001em9GY2R1VVFSZkJJXC9YMnVrd1BLU0s1V3VrZVpKbUExUGFybmRBTDlSIiwibWFjIjoiNTA0NzQ2ZjM5YjY0YzA0ZjA
  2MjlmNmIzNWU0OGQ5MTEwMmYyOTlkOGI5M2U0YWE2ZmExNjc0MDlkMDAxYjRhOCJ9; expires=Sun, 05-Jan-2020 04:36:02 GM
  T; Max-Age=86400; path=/; httponly
  Transfer-Encoding: chunked
```

Figure 10. HTTP Header.

Table 2. List of Know Vulnerability.

Service	Vulnerability
Bootstrap 3.3.7	CVE-2019-8331
AdminLTE 2.4.2	XSS on Morris.js
Bootstrap3 Wysihtml5 Bower 2014-09-26	XSS onload event
Datatables 1.10.16	CVE-2015-6584
JQuery 3.3.1	Prototype Pollution
JQuery UI 1.11.4	XSS on Dialog Class
Tiny MCE 4.17	XSS on Xlink:href

Table 2 is the list of known vulnerability on SIMAK-NG Application.

4.10. Map Application Architecture (OTG-INFO-010)

SIMAK-NG Application can be access if user have authentication from Single Sign On (SSO) portal that is IMISSU (Intergrated Management Information System, the Strategic of UNUD). On OTG-INFO-004 we know any port that used on SIMAK-NG Application.

Therefore we can make assumption that SIMAK-NG Application architecture will seem like figure 11.

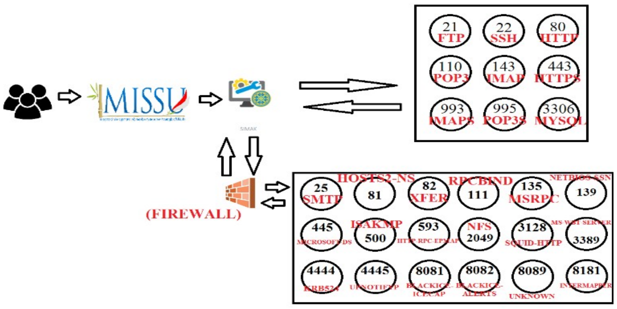


Figure 11. SIMAK-NG Application Architecture.

Figure 10 is the SIMAK-NG Application Architecture assumption from OTG-INFO-004 and authentication process.

5. Conclusion

The final result of this testing represented as 0,1, or -1 state for each testing. That represented on table 3.

Table 3. Test Result.

Testing	Result
Conduct Search engine discovery/ reconnaissance for information leakage (OTG-INFO-001)	0
Fingerprint Web Server (OTG-INFO-002)	1
Review Webserver Metafiles for Information Leakage (OTG-INFO-003)	0
Enumerate Applications on Webserver (OTG-INFO-004)	1
Review webpage comments and metadata for information leakage (OTG-INFO-005)	0
Identify application entry points (OTG-INFO-006)	1
Map execution paths through application (OTG-INFO-007)	1
Fingerprint Web Application Framework (OTG-INFO-008)	1
Fingerprint Web Application (OTG-INFO-009)	1
Map Application Architecture (OTG-INFO-010)	1

OTG-INFO-001 testing has been performed, but there is no any sensitive information found that’s why the result is 0. From OTG-INFO-002 we know about web server used is nginx/ 1.14.0 (ubuntu), the result for this testing is 1.

The result of OTG-INFO-003 is there is no sensitive information found from metafiles, the result for this testing is 0. OTG-INFO-004 testing has been performed and we know any port used on SIMAK-NG Application, the result for this testing is 1.

OTG-INFO-005 testing has been performed and there is no sensitive information from HTML Source code. The result of OTG-INFO-006 is we can know any entry point on SIMAK-NG application, so the result is 1.

The result of OTG-INFO-007 is we know get method path from SIMAK-NG Application, so the result for this testing is 1. OTG-INFO-008 testing has been performed and we know web application framework used is Laravel, the result for this testing is 1.

OTG-INFO-009 testing has been performed and we know there is some service that used have known vulnerability, the result for this testing is 1. We can make assumption about SIMAK-NG Application (as a black-box pentester) that’s why the result for OTG-INFO-010 is 1.

There is 10 testing has been performed and 3 testing get negative result. From all of this we know what is the next step we can do to perform penetration testing, like the pentest method has been explain before on figure 2.

For example from OTG-INFO-004 we know there is no LDAP protocol open, so OTG-INPVAL-006 cant be perform. We know too there is mysql port open, so OTG-INPVAL-005 can be perform.

References

[1] M. Felderer, M. Buchler, M. Johns, A. D. Brucker, R. Breu, and ¨ A. Pretschner, “Security testing: A survey,” in Advances in Computers. Elsevier, 2016, vol. 101, pp. 1–51.

[2] I Putu Agus Eka Pratama, Anak Agung Bagus Arya Wiradarma, "Open Source Intelligence Testing Using the OWASP Version 4 Framework at the Information Gathering Stage (Case Study: X Company)", International Journal of Computer Network and Information Security(IJCNIS), Vol.11, No.7, pp.8-12, 2019.DOI: 10.5815/ijcnis.2019.07.02

[3] Abel Yeboah-Ofori, P. A. B. (2017). "Cyber Intelligence and OSINT: Developing Mitigation Techniques Against Cybercrime Threats on Social Media." International Journal of Cyber-Security and Digital Forensics 7(1): 11.

[4] Young B. Choi and Kenneth P. LaCroix, “Building a Penetration Testing Device for Black Box using Modified Linux for Under $50” International Journal of Advanced Computer Science and Applications(IJACSA), 8(1), 2017. http://dx.doi.org/10.14569/IJACSA.2017.080103

[5] Bahrun Ghozali, K., Sudarmawan and (2018). "Mendeteksi Kerentanan Keamanan Aplikasi Website Menggunakan Metode Owasp (Open Web Application Security Project) untuk Penilaian Risk Rating "Creative Information Technology Journal 4(4): 11

[6] Raden Teduh Dirgahayu, Y. P., Adi Fajaryanto (2015). "Penerapan Metode ISSAF dan OWASP versi 4 Untuk Uji Kerentanan Web Server "Jurnal Imiah NERO 1(3): 7.

[7] Yunanri W, I. R., Anton Yudhana (2018). "Analisis Deteksi Vulnerability Pada Webserver Open Jurnal System Menggunakan OWASP Scanner." Jurnal Rekayasa Teknologi Informasi 2(1): 8.

[8] The OWASP Foundation, “ZAP Proxy.”

[9] I. Riadi, R. Umar, and W. Sukarno, “Vulnerability of Injection Attacks Against The Application Security of Framework Based Bebsites Open Web Access Security Project (OWASP),” J. Inform., vol. 12, no. 2, pp. 53–57, 2018.

[10] NMAP.Org, Introduction, Retrieved 2013. Diakses tanggal 3 Juni 2020