Jurnal Informasi dan Teknologi

http://www.jidt.org

2022 Vol. 4 No. 1 Hal: 70-75 e-ISSN: 2714-9730

Pengujian dan Analisis Keamanan Website Menggunakan Acunetix Vulnerability Scanner

Afif Zirwan[1]✉

[1]Institut Teknologi Padang

afifzirwan47@gmail.com

Abstract

The Covid-19 pandemic has indirectly accelerated the Industrial Revolution 4.0 where technology has become an important point in industrial movement. Changes that are forced to be fast, sometimes escape some application developers to test the product to be implemented, because they are required to quickly compensate for these changes. The Padang Institute of Technology (ITP) also made rapid changes to its official website, to provide actual information on campus activities during this pandemic. This study aims to test and analyze the extent of the security of the ITP website and provide troubleshooting suggestions from the results of the analysis. Testing is done using the Acunetix Vulnerability Scanner tool. The method used is descriptive analysis, namely the data obtained are presented in the form of sentences that are described, so as to provide clarity from the results of the analysis carried out. From the data obtained, the ITP website is at threat level 3 which is included in the High category. In this study there were 714 alerts or gaps found consisting of 94 at the high level, 25 at the medium level, 46 at the low level and 549 at the informational level. Based on the analysis, improvement and testing carried out in this research on the ITP website, the resulting threat level is already at level 1, which can be concluded that the ITP website is classified as safe from security holes.

Keywords: Revolution 4.0, Security, Algorithms, Infrastructure, Acunetix Vulnerability Scanner.

Abstrak

Pandemi Covid-19 secara tidak langsung mempercepat Revolusi Industri 4.0 dimana teknologi menjadi poin penting dalam pergerakan industri. Perubahan yang dipaksa cepat, kadang kala meluputkan beberapa pengembang aplikasi untuk melakukan pengujian terhadap produk yang akan diimplementasikan, karena dituntut cepat untuk mengimbangi perubahan tersebut. Institut Teknologi Padang (ITP) juga melalukan perubahan cepat terhadap *website* resminya, untuk menyediakan informasi aktual dari kegiatan kampus dimasa pandemi ini. Penelitian ini bertujuan untuk melakukan pengujian dan analisa sejauh mana keamanan *website* ITP dan memberikan saran pemecahan masalah dari hasil analisa. Pengujian dilakukan dengan menggunakan *tools* Acunetix Vulnerability Scanner. Metode yang digunakan adalah analisa deskriptif, yaitu data yang diperoleh di sajikan dalam bentuk kalimat yang dideskripsikan, sehingga memberikan kejelasan dari hasil analisa yang dilakukan. Dari data yang diperoleh, website ITP berada pada *threat level* 3 yang termasuk ketegori *High*. Pada penelitian ini terdapat 714 *alert* atau celah yang ditemukan yang terdiri dari 94 pada *level high*, 25 pada *level medium*, 46 pada *level low* dan 549 pada *level informational*. Berdasarkan analisa, perbaikan dan pengujian yang dilakukan pada penelitian ini terhadap *website* ITP, menghasilkan *threat level* sudah pada *level* 1, yang dapat disimpulkan *website* ITP sudah tergolong aman dari celah keamanan.

Kata kunci: Revolusi 4.0, Keamanan, Algoritma, Infrastruktur, Acunetix Vulnerability Scanner.

1. Pendahuluan

Industri 4.0 atau Revolusi Industri keempat merupakan istilah yang umum digunakan untuk tingkatan perkembangan industri teknologi di dunia. Pada tingkatan keempat ini, dunia memang fokus kepada teknologi-teknologi yang bersifat digital, oleh karena itu teknologi informasi sangat-sangat menjadi tumpuan untuk industri yang bertujuan untuk mempermudah dan mempercepat proses-proses untuk membuat produk.

Pandemi Covid-19 secara tidak lansung mempercepat Revolusi Industri 4.0, dimana banyak kegitan dilakukan tidak dengan bertatap muka secara lansung, melainkan melalui pertemuan *online* melalui *gadget* masing-masing. Begitu juga dengan pencarian informasi, semua dilakukan dengan cepat melalui internet. Dengan faktor tersebut, banyak industri dan instansi berpacu untuk melakukan pembaharuan terhadap layanan informasi mereka yang agar pengiriman data dan informasi meningkat. Disamping keuntungan tersebut, tingkat resiko dan ancaman penyalahgunaan teknologi informasi juga menjadi semakin meningkat [1].

Perubahan yang sangat cepat, kadang kala meluputkan *developer* dalam melakukan pengujian terhadap aplikasi yang dibangun. Pengujian merupakan proses yang sangat penting didalam pengembangan perangkat lunak yang berkualitas tinggi, karena dari beberapa kesalahan yang dianggap tidak penting beresiko sangat berbahaya hal ini menjadi celah (*vulnerability*) bagi *attacker* untuk memanfaatkan informasi yang di curi

Diterima: 01-03-2022 | Revisi: 18-03-2022 | Diterbitkan: 31-03-2022| doi: 10.37034/jidt.v4i1.190

melalui serangan kepada aplikasi. Kebutuhan akan *vulnerability assessment* selama ini biasanya dipandang sebelah mata, karen hanya dianggap sebagai kegiatan formalitas dan sedikit orang yang melakukan kegiatan ini [2]. Salah satu sistem yang umumnya menjadi sasaran *hacker* dan *cracker* adalah aplikasi berbasis *website*. Hal tersebut dikarenakan pemanfaatan aplikasi mengalami pertumbuhan yang sangat pesat saat ini [3]. Serangan yang dilakukan dapat berupa *Cross Site Scripting* (XSS), *Cross Site Request Forgeri* (CSRF), SQL *injection* dan lain sebagainya [4]. Serangan SQL *injection* merupakan sebuah aksi *hacking* yang dilakukan di aplikasi *client* dengan cara memodifikasi perintah SQL yang ada di memori aplikasi *client* dan mengekploitasi aplikasi menggunakanh basis data untuk penyimpanan data [5]. Penelitian lainya juga mendapatkan hasil jenis serangan seperti *Local File Inclusion* (LFI) dan parameter *tampering* [6].

Celah keamanan (*vulnerability*) sistem jaringan komputer merupakan sebuah kelemahan, kekurangan atau celah pada sistem, yang dapat dimanfaatkan oleh satu atau lebih dari penyerang untuk melakukan serangan yang dapat membahayakan kerahasiaan, integritas, atau ketersediaan suatu system [7]. *Vulnerability* adalah suatu kelemahan yang mengancam nilai *integrity*, *confidentiality*, dan *availability* dari suatu asset [8]. *Vulnerability assessment* merupakan bagian dari *risk assessment* yang terdiri dari *risk analysis*, *policy development*, *training and implementation*, dan *vulnerability assessment* and *penetration testing* [9]. *Vulnerability Assessment* (VA) adalah proses pemindaian sistem atau *software* dan jaringan untuk mengetahui kelemahan dan celah yang ada, celah ini memberikan *backdoor* ke penyerang untuk menyerang korban. sistem memiliki *access control vulnerability*, *boundary condition vulnerability*, input *validation vulnerability*, *authentication vulnerabilities*, *configuration weakness vulnerabilities*, *dan exception handling vulnerabilities* [2].

Dalam tataran konsep proses VA merupakan proses yang sangat kompleks karena melibatkan seluruh komponen dalam TI. Dalam tataran teknis pun merupakan proses yang beresiko mengingat adanya peluang untuk merusak atau mengganggu kinerja sistem yang berlangsung. Proses VA secara garis besar dapat dibagi dalam tiga tahapan [10]:

a. Penentuan batasan proyek

b. Pelaksanaan assessment

c. Pelaporan akhir

Vulnerability testing banyak digunakan untuk meningkatkan kesadaran tentang pentingnya keamanan informasi [11]. Penilaian kerentanan bisa mendeteksi hampir semua celah kerentanan yang biasanya terjadi pada sebuah system [12]. Beberapa faktor dari dari celah keamanan bisa terjadi karena kurangnya sistem pengamanan *website* dan kekeliruan programmer ketika melakukan *coding* [13].

Salah satu cara untuk melakukan evaluasi keamanan *website* menggunakan perangkat lunak yang khusus dirancang untuk mengetahui kerentanan yang ada pada suatu sistem yaitu Acunetix Vulnerability Scanner [14] Pentest-tools.com, *vulnerability scanner*, OWASP ZAP [8]. Pengujian ini juga tidak terbatas pada aplikasi yang di kustom sendiri, aplikai CMS (*Content Management System*) seperti OJS juga menjadi target uji [9]. Pengujian *vulnerability* dilakukan untuk pengukuran atau *assessment* yang mutlak dilakukan untuk mendapatkan peningkatan kualitas dan salah satu cara pengukuran terhadap keamanan sistem. Hasil dari assesment menjadi bahan pertimbangan bagi *developer* untuk mengambil tindakan pencegahan dan mengetahui cara kerja dari *attackers* [15].

Begitu juga dengan Institut Teknologi Padang (ITP), baru saja meluncurkan *website* resmi versi ke 3. Alasan ITP meluncurkan versi ialah untuk menutupi celah keamaan yang terjadi pada versi 2 dimana pada versi ini masih menggunakan teknolgi scripting yang masih lama dimana rentan menjadi target serangan oleh *attacker*. Oleh karena itu pengujian *vulnerability* penting dilakukan, yang hasilnya bukan menggaransi sistem akan bebas dari resiko serangan, tetapi dapat meminimalisir serangan yang dapat disalahgunakan, karena untuk menjelajahi semua aspek harus dilakukan pengujian tingkat lanjut [16].

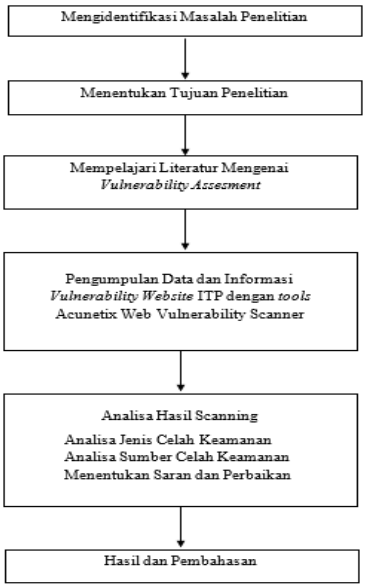
Berdasarkan paparan diatas, penulis ingin melalukan analisa dan pengujian serta terhadap versi 3 dari *website* ini, dengan tujuan agar dapat memberikan saran perbaikan dan peningkatan dari kemanan *website* ini. Berdasarkan uraian diatas, penulis mengangkat permasalahan diatas sebagai judul penelitian yang berjudul "Pengujian dan Analisis Kemanan *Website* Menggunakan Acunetix Vulnerability Scanner".

2. Metodologi Penelitian

Metodologi penelitian mengidentifikasi semua tahapan yang digunakan dalam pembuatan struktur kerja atau biasa dikenal dengan kerangka kerja. Kerangka kerja digunakan untuk membuat tahapan – tahapan yang akan diselesaikan dalam penelitian, sehingga tahapan tersebut mempengaruhi setiap tahapan dalam mencapai tujuan penelitian. Tujuan penelitian ini adalah untuk menganalisis *vulnerability* yang terdapat pada *website* Institut Teknologi Padang dengan menggunakan *tools* Acunetix Web Vulnerability Scanner. Acunetix ini adalah salah satu program paling sukses di pasar untuk mendeteksi celah keamanan SQL *injection* dan XSS [17]. Data yang sudah diperoleh akan ditelaah celah keamanan yang ditemukan satu persatu berdasarkan jenis celah keamanan. Dengan mengelompokkan jenis celah keamanan memudahkan untuk menganalisi. Informasi ini dijadikan landasan untuk mengetahui apa

yang menjadi penyebab dan pemberian solusi untuk masalah celah keamanan ini.

Kerangka kerja penelitian adalah suatu tahapan dalam menyelesaikan suatu permasalahan yang akan diteliti. Kerangka kerja disajikan pada Gambar 1.



Gambar 1. Kerangka Kerja Penelitian

Kerangka kerja pada Gamabr 1 merupakan penggambaran terperinci yang didasari dengan cara yang terorganisir sehingga penelitian dapat mencapai tujuannya dan memilki penilaian yang semestinya.

2.1. Mengidentifikasi Masalah Penelitian

Identifikasi masalah adalah tahapan untuk menemukan masalah sebelum melakukan penelitian. Permasalahan yang dihadapi adalah *vulnerability assesment* yang seharusnya dilakukan sebelum *release* dianggap tidak terlalu penting dan hanya berpedoman pada *black box test* atau uji fungsionalitas sistem. Pada *website* ITP belum pernah dilakukan *vulnerability assesment* sama sekali, yang dimana kerentanan pada keamanan *website* merupakan hal yang harus diperhatikan bagi setiap institusi agar terhindar dari tindakan kejahatan di dunia maya.

2.2. Menetapkan Tujuan Penelitian

Tahapan penentuan tujuan ini merupakan tahapan di mana peneliti mengemukakan tujuan dari penelitian agar tidak keluar dari hasil yang ingin diperoleh. Tujuan penelitian ini adalah untuk menganalisa *vulnerability* atau celah keamanan yang ada pada *website* ITP dan memberikan solusi terhadap masalah yang ditemukan, dengan tujuan laporan dari hasil penelitian ini dapat menjadi acuan bagi pengembang atau administrator sistem untuk melakukan perbaikan dan pengembangan sistem.

2.3. Mempelajari Literatur Mengenai *Vulnerability Assesment*

Mempelajari literatur merupakan fase dalam mencari tahu tentang *vulnerabilty* dan *vulnerabilty assesment*. Sebagai metode yang digunakan dalam penelitian, akan digunakan untuk menganalisis *vulnerability* yang terdapat pada *website* ITP, literatur yang sudah didapatkan akan dipilih dan dicocokan dengan apa aja yang akan dipakai di dalam penelitian. Sumber dari literatur ini dapat berasal dari artikel, jurnal ilmiah mengenai *vulnerability assesment*, serta referensi-referensi yang berhubungan dengan penelitian.

2.4. Pengumpulan Data dan Informasi *Vulnerability Website* ITP

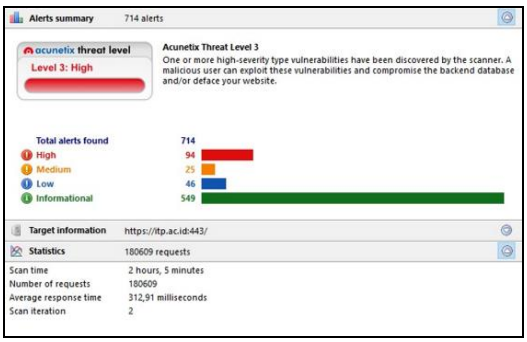
Tujuan dari pengumpulan data adalah untuk mendapatkan suatu informasi dari data-data yang dibutuhkan untuk penelitian agar mencapai tujuan yang diharapkan. Pada tahapan ini dijelaskan berupa kegiatan *scanning* dari *website* ITP dengan menggunakan *tools* Acunetix web *vulnerability scanner*. Dari data yang didapat akan dihimpun dalam bentuk tabulasi agar mudah dilakukan analis.

2.5. Analisa Hasil *Scanning*

Pada Tahapan ini, data yang didapatkan dari hasil *scanning* akan ditelaah celah keamanan yang ditemukan satu berdasarkan jenis celah keamanan. Dengan mengelompokkan jenis celah keamanan memudahkan untuk menganalisis. Data yang didapat dari proses scanning menggunakan Acunetix web *vulnerability scanner* akan diuraikan dengan tujuan data tersebut dapat menjadi informasi atau *report* yang dapat dijadikan saran perbaikan untuk pengelola *website*. Analisa yang dilakukan adalah dengan menjelaskan jenis celah keamanan, sumber celah, solusi perbaikan dari celah kemanan tersebut. Selain dengan menggunakan metode tabulasi, tata cara penulisan *report* harus disampaikan dengan bahasa yang teknis, kenapa demikian karena hasil dari *report* ini akan diberikan kepada pengembang atau *developer*.

3. Hasil dan Pembahasan

Dari hasil *scanning* yang dilakukan, *website* ITP mendapatkan hasil *score* pada level 3, yang berarti *website* ini tidak dalam kondisi aman. Durasi *scan* yaitu 2 jam 5 menit dengan iterasi *scan* atau perulanghan *scan* dilakukan sebanyak 2 kali. Pada penelitian ini terdapat 714 *alert* atau celah yang ditemukan yang terdiri dari 94 pada *level high*, 25 pada *level medium*, 46 pada *level low* dan 549 pada *level informational*. Pada penilitian ini, target yang dicapai adalah perbaikan sampai menjadi *level* 1. Oleh karena itu pada penjelasan saran perbaikan hanya di fokuskan pada *alert high* dan *medium* yang disajikan pada Gambar 2.



Gambar 2. Hasil Scanning *Website*

Mengacu pada batasan penelitian ini yaitu memberikan *report* dan perbaikan pada aplikasi yang di uji coba, maka dilakukan perbaikan aplikasi hingga *threat level* menjadi *level* 1.

a. *Cross Site Scripting* (XSS)

Untuk menangani permasalahan XSS maka input dari semua *method* dilakukan *filtering* dengan fungsi yang disajikan pada Gamabr 3.

```
$this->security->xss_clean(preg_replace("/[^a-zA-Z0-9\ ]/", "",$VariableInput));
```

Gambar 3. Fungsi Filtering XSS

Script pada Gambar 3 digunakan fungsi xss_clean yang merupakan bawaan dari *framework* CodeIgniter untuk memfilter karakter yang di input. Sedangkan fungsi preg_replace digunakan untuk fungsi REGEX atau *regular expression* membatasi jenis karakter yang bisa diterima dari inputan. Pada kasus ini REGEX yang digunakan adalah hanya bisa menginputkan karakter a-z, A-Z dan 0-9. Disisi konfigurasi aplikasi cukup dengan menghidupkan config global xss *filtering* dengan *true* pada *file* config.php yang disajikan pada Gambar 4.

```
$config['global xss filtering'] = true;
```

Gambar 4. Global XSS Filtering

Pada *header* halaman tempat lokasi yang di *inject* XSS, terdapat kesalahan *scripting* dimana untuk mengambil parameter url tidak perlu di gunakan *script* $_SERVER['REQUEST_URI'] yang akan menangkap semua inputan *method* GET pada url yang menyebabkan semua inputan tersebut ter-*inject* ke HTML. Seharus nya untuk mendapatkan *url* aktif cukup dengan fungsi current_url() seperti Gambar 5 .

```
content="https://<?= $_SERVER['SERVER_NAME'].$_SERVER['REQUEST_URI'] ?>">

content="<?= current_url() ?>">
```

Gambar 5. Penggantian Metode Penulisan Script URL

b. *Application Error Message*

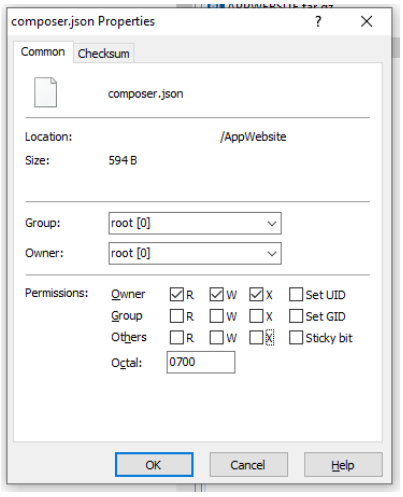
Untuk menangani kasus ini dilakukan perbaikan pada *query* yang terdapat *error*, dan untuk pencegahannya, fungsi dari error_reporting dijadikan 0 atau pada kasus ini *settingan* pada konfigurasi *environment* menjadi *production* seperti *script* pada Gambar 6.

```
define('ENVIRONMENT', isset($_SERVER['CI_ENV']) ? $_SERVER['CI_ENV'] :
'production');
 switch (ENVIRONMENT) {
    case 'development':
        error_reporting(-1);
        ini_set('display_errors', 1);
        break;
    case 'testing':   case 'production':
        ini_set('display_errors', 0);
        if (version_compare(PHP_VERSION, '5.3', '>=')) {
            error_reporting(E_ALL & ~E_NOTICE & ~E_DEPRECATED &
~E_STRICT & ~E_USER_NOTICE & ~E_USER_DEPRECATED);
        } else {
            error_reporting(E_ALL & ~E_NOTICE & ~E_STRICT &
~E_USER_NOTICE);
        }
        break;
    default:
        header('HTTP/1.1 503 Service Unavailable.', TRUE, 503);
        echo 'The application environment is not set correctly.';
        exit(1); // EXIT_ERROR
}
```

Gambar 6. Konfigurasi Environment Aplikasi

c. *Development Configuration File*

Pada kasus ini ditemukan *file* konfigurasi yaitu composesr.json yang dapat diakses oleh *user* secara gamblang. Untuk menyelesaikan permasalahan ini dapat diberikan pengaturan hak akses *file* dengan oktal 700 yang berfungsi *file* dapat diakses ditulis dan dieksekusi hanya oleh *owner* saja. Pengaturan disajikan pada Gamabr 8.



Gambar 7. Pengaturan Hak Akses File

d. *Error Message Page*

Bug yang terdapat pada kasus ini adalah tidak adanya fungsi *index* pada *controller* ini yang menyebabkan

jika *file* ini diakses secara lansung akan menampilkan pesan *eror*. Untuk menangani kasus ini dilakukan penambahan fungsi *controller index* dan diarahkan kehalaman 404. Untuk mencegan pesan *error* tampil maka sama dengan kasus *applikcation error message* dengan cukup mengatur fungsi dari error_reporting() dijadikan 0 atau pada kasus ini settingan pada konfigurasi *environment* menjadi *production* seperti *script* pada Gambar 8.

```
28      public function index()
29      {
30          show_404();
31      }
```

Gambar 8. Penambahan *Controller Index*

e. *Host Header Attack*

Host header attack yang terjadi pada kasus ini merupakan kasus yang cukup unik, kenapa demikian dari pentest yang dilakukan ketika *host header* nya dirubah salah satu *controller* pada *website* ini diarahkan ke ke aplikasi lain yang memiliki *controller* aplikasi yang sama URL nya, oleh karena itu perbaikan pada kasus ini juga di lakukan pada dua aplikasi ini. Pada kasus ini dibutuhkan konfigurasi pada sisi *webserver* dan ada sisi *script*. Berikut konfigurasi yang dilakukan yang disajikan pada Gambar 9.

```
server_name itp.ac.id www.itp.ac.id; # Server name disesuaikan

if ( $host !~* ^(itp.ac.id|www.itp.ac.id)$ ) {
    return 444;
 }
 if ( $http_host !~* ^(itp.ac.id|www.itp.ac.id)$ ) {
    return 444;
 }
```

Gambar 9. Konfigurasi *Header* pada Nginx

Pada Gambar 9 disajikan *host* yang melakukan *request* tidak sama dengan itp.ac.id maka *webservice* NGINX akan lansung me-*redirect* ke halaman ke 444 yang berarti NGINX akan menutup atau memutuskan respon koneksi ke *client*. Kode 444 ini banyak di gunakan pada NGINX untuk membalas respon dari *request* yang berpotensi sebagai *malicious or malformed requests*. Pada sisi aplikasi, juga diberikan *script* yang sama tetapi *return* yang di berikan adalah *redirect* ke 404 seperti Gambar 10.

```
if (base_url() != "https://itp.ac.id/") {
    show_404();
}
$domains = ['itp.ac.id'];
if (!in_array($_SERVER['SERVER_NAME'], $domains)) {
    show_404();
}
```

Gambar 10. Konfigurasi Header Pada Aplikasi

f. HTML *From Without* CSRF *Protection*

Pada permasalahan ini metode *form* yang digunakan adalah *method* GET yang dimana *form* tersebut tidak bisa diamankan dengan dari CSRF, oleh karena itu *form* tersebut diganti dengan *method* POST dan diaktifkan CSRF dengan mengirimkan *key* atau token. Dengan adanya *key* atau token ini, maka setiap *form* akan memiliki validasi tersendiri ketika dilakukan POST. Konfigurasi yang dilakukan adalah pada *file config* dari aplikasi seperti Gambar 11.

```
$config['csrf_protection']    = true;
$config['csrf_token_name']    = 'Token';
$config['csrf_cookie_name']   = 'CToken';
$config['csrf_expire']        = 7200;
$config['csrf_regenerate']    = true;
```

Gambar 11. Konfigurasi CSRF Protection

Sedangkan pada sisi *script* program, pemanggilan *form* dirubah yang dari biasanya di tuliskan dengan tag *form* diganti dengan fungsi form_open(), yang dimana jika tag ini di *echo* akan menyisipkan token dari *form* tersebut yang disajikan pada Gamabr 12 dan Gambar 13.

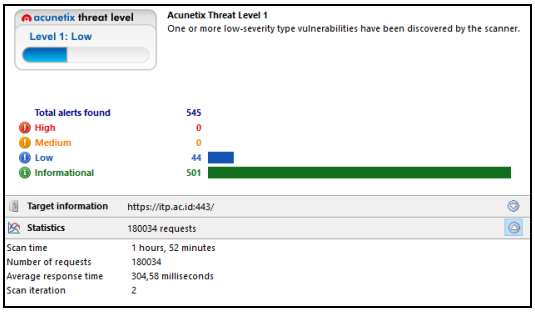
```
<div class="col-lg-12 mt-5">
    <?php echo form_open(base_url('search'), ' name="search" method="POST"') ?>
    <div class="search-wrapper active">
        <div class="input-holder">
            <input type="text" name="keywords" class="search-input" value="<?= $_SES
            <button class="search-icon"><span></span></button>
        </div>
        <!-- <span class="close" onclick="searchToggle(this, event);"></span> -->
    </div>
    <?php form_close(); ?>
```

Gambar 12. *Syntax* HTML *From* dengan CSRF *Protection*

```
<form action="https://itp.ac.id/search"  name="search" method="POST"
dden" name="Token" value="b510bcadee2cb7aecef420c531b88273" />
<div class="search-wrapper active">
    <div class="input-holder">
        <input type="text" name="keywords" class="search-input" valu
        <button class="search-icon"><span></span></button>
    </div>
    <!-- <span class="close" onclick="searchToggle(this, event);"></
</div>
</form>            </div>
```

Gambar 13. HTML *From* dengan CSRF *Protection*

Setelah kegiatan perbaikan yang dilakukan, maka dilakukan *scanning* ulang yang berguna memastikan apakah perbaikan yang dimaksud sudah berfungsi dan menutupi celah keamanan yang terdeteksi pada *scanning* awal. Pada *scanning* ulang ini dilakukan selama 1 jam 52 menit dengan iterasi *scan* sebanyak 2 kali. Hasil *scan* disajikan pada Gamabr 14.



Gambar 13. Hasil Scanning Setelah Perbaikan

Hasil yang diperoleh adalah *website* ITP sudah mendapatkan peringkat 1 atau *low threat* level yang dimana ini sudah dikategoringan aman berdasarkan Acunetix WVS karena pada level 1 *vulnerability* yang ditemukan hanya berifat informasional saja.

4. Kesimpulan

Berdasarkan pengujian dan analisa *vulnerability* pada website ITP, *website* ITP mendapatkan *threat* pada *level* 3 yang termasuk ketegori *high* dengan 2 kali iterasi *scanning*. Setelah dilakukan analisa dan perbaikan pada *website* ITP, dilakukan pengujian dengan *scanning* ulang sebanyak 2 kali, yang dimana hasil dari perbaikan ini *website* ITP mendapatkan *threat* pada *level* 1 yang termasuk ketegori *low*. Menggunakan *tools* Acunetix WVS dapat membantu *assesment* dan perbaikan untuk mengurangi *vulnerability* yang ditemukan.

Daftar Rujukan

[1]. A. Bustami and S. Bahri, "Ancaman, Serangan dan Tindakan Perlindungan pada Keamanan Jaringan atau Sistem Informasi: Systematic Review," 2020. doi: http://dx.doi.org/10.33592/unistek.v7i2.645.

[2]. J. N. Goel and B. M. Mehtre, "Vulnerability Assessment & Penetration Testing as a Cyber Defence Technology," Procedia Comput. Sci., vol. 57, pp. 710–715, 2015, doi: http://dx.doi.org/ 10.1016/j.procs.2015.07.458

[3]. F. Al Fajar, "Analisis Keamanan Aplikasi Web Prodi Teknik Informatika Uika Menggunakan Acunetix Web Vulnerability," *Inova-Tif*, vol. 3, no. 2, p. 110, 2020, doi: http://doi.org/10.32832/inova-tif.v3i2.4127

[4]. I. Riadi, A. Yudhana, and Y. W, "Analisis Keamanan Website Open Journal System Menggunakan Metode Vulnerability Assessment," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 7, no. 4, p. 853, 2020, doi: http://dx.doi.org/10.25126/jtiik.2020701928

[5]. Ade Bastian, Harun Sujadi, and Latiful Abror, "ANALISIS KEAMANAN APLIKASI DATA POKOK PENDIDIKAN (DAPODIK) MENGGUNAKAN PENETRATION TESTING DAN SQL INJECTION," *INFOTECH J.*, vol. 6, pp. 65–70, 2020, doi: http://dx.doi.org/10.31949/infotech.v6i2.848

[6]. U. Gupta, S. Raina, P. Verma, P. Singh, and M. M. Aggarwal, "Web Penetration Testing," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 8, no. 5, pp. 56–60, 2020, doi: http://doi.org/10.22214/ijraset.2020.5011

[7]. B. W. Retna Mulya and A. Tarigan, "Pemeringkatan Risiko Keamanan Sistem Jaringan Komputer Politeknik Kota Malang Menggunakan Cvss Dan Fmea," *Ilk. J. Ilm.*, vol. 10, no. 2, pp. 190–200, 2018, doi: http://doi.org/10.33096/ilkom.v10i2.311.190-200.

[8]. E. Irawadi Alwi and F. Umar, "Analisis Keamanan Website Menggunakan Teknik Footprinting dan Vulnerability Scanning," 2020. doi: https://doi.org/10.19184/isj.v5i2.18941

[9]. F. Wibowo and A. Purwo Wicaksono, "Uji Vulnerability pada Website Jurnal Ilmiah Universitas Muhammadiyah Purwokerto Menggunakan OpenVAS dan Acunetix WVS," *J. Inform.*, vol. 6, no. 2, pp. 212–218, 2019, doi: https://doi.org/10.31294/ji.v6i2.5925

[10]. A. Priandoyo, "Vulnerability Assessment untuk Meningkatkan Kesadaran Pentingnya Keamanan Informasi," *J. Tek. Inform. dan Sist. Inf.*, vol. 1, no. 2, pp. 73–83, 2006.

[11]. I. Riadi, Herman, and A. Z. Ifani, "Optimasi Keamanan Web Server terhadap Serangan Broken Authentication Menggunakan Teknologi Blockchain," *JISKA (Jurnal Inform. Sunan Kalijaga)*, vol. 6, no. 3, pp. 139–148, 2021, doi: https://doi.org/10.14421/jiska.2021.6.3.139-148

[12]. Y. A. Pohan, "Meningkatkan Keamanan Webserver Aplikasi Pelaporan Pajak Daerah Menggunakan Metode Penetration Testing Execution Standar," *J. Sistim Inf. dan Teknol.*, vol. 3, pp. 1–6, 2021, doi: https://doi.org/10.37034/jsisfotek.v3i1.36

[13]. I. Syarifudin, "Pentesting dan Analisis Keamanan Web Paud Dikmas," *Pentesting Dan Anal. Keamanan Web Paud Dikmas*, no. April, p. 2, 2018, doi: https://doi.org/10.5281/zenodo.1211847

[14]. R. Mayasari, A. A. Ridha, D. Juardi, and K. A. Baihaqi, "Analisis Vulnerability pada Website Universitas Singaperbangsa Karawang menggunakan Acunetix Vulnerability," 2020. doi: https://doi.org/10.35706/sys.v2i1.3450

[15]. M. Orisa and M. Ardita, "VULNERABILITY ASSESMENT UNTUK MENINGKATKAN KUALITAS KEMANAN WEB," 2021. doi: https://doi.org/10.36040/mnemonic.v4i1.3213

[16]. G. S. T and Sasikala D, "Vulnerability Assessment of Web Applications using Penetration Testing," *Int. J. Recent Technol. Eng.*, vol. 8, no. 4, pp. 1552–1556, Nov. 2019, doi: https://doi.org/10.35940/ijrte.b2133.118419.

[17]. M. Ula, "Evaluasi Kinerja Software Web Penetration Testing," *TECHSI - J. Tek. Inform.*, vol. 11, no. 3, p. 336, 2019, doi: https://doi.org/10.29103/techsi.v11i3.1996.