JITKOM

Jurnal Ilmu Teknik dan Komputer
Vol. 07. No. 01, Januari 2023: 35-43
p-ISSN: 2548-740X e-ISSN: 2621-1491

Uji Penetrasi *Server* Universitas PQR Menggunakan Metode *National Institute Of Standards And Technology* (NIST SP 800-115)

Syifa Sabrina Anelia*, Jayanta dan Bayu Hananto

Informatika, Universitas Pembangunan Nasional Veteran, Jakarta
*syifasabrinaa@upnvj.ac.id

***Abstrak*— Ancaman keamanan serangan siber terjadi di beberapa universitas. Data penting yang terletak pada server organisasi bisa saja diretas oleh orang yang tidak berhak. Salah satu cara menghindari peretasan adalah menutup celah-celah keamanan yang dimiliki sistem. Sebelum menutup celah keamanan, tentu harus diketahui celah keamanannya, dengan melakukan pengujian seperti yang dilakukan oleh peretas, namun dengan prosedur yang telah disetujui. Pada penelitian ini dilakukan pengujian penetrasi yang bertujuan menguji kerentanan serta menemukan celah keamanan yang ada pada server universitas, sehingga nantinya dapat ditangani dengan baik oleh Universitas PQR. Pengujian menggunakan metode National Institute of Standards and Technology (NIST SP 800-115) yang terdiri dari 4 fase pengujian, yaitu planning, discovery, attack, dan reporting. Hasil yang didapatkan pada penelitian ini yaitu ditemukannya 13 kerentanan yang dapat dieksploitasi dengan rincian 2 kerentanan termasuk kategori critical yaitu Default Credentials dan PHP Unsupported Version Detection, 3 kerentanan termasuk kategori high yaitu SSL Version 2 and 3 Protocol Detection, PHP < 7.3.24 Multiple Vulnerabilities, SSL Medium Strength Cipher Suites Supported (SWEET32), 8 kerentanan termasuk kategori medium yaitu SSL Certificate Cannot Be Trusted, SSL Self-Signed Certificate, TLS Version 1.0 Protocol Detection, PHPinfo() Information Disclosure, Unencrypted Password Form, HTTP TRACE / TRACK Methods Allowed, SSL Certificate Expiry, SSL RC4 Cipher Suites Supported (Bar Mitzvah), dan 1 kerentanan adalah false positive yaitu PHP < 7.1.33 / 7.2.x < 7.2.24 / 7.3.x < 7.3.11 Remote Code Execution Vulnerability. Hasil pengujian menunjukkan bahwa server universitas masih rentan, sehingga perlu penanganan dan perbaikan kerentanannya oleh pihak Universitas PQR.**

Kata Kunci*— *Keamanan Data, NIST SP 800-115, Protocol, Siber, Uji Penetrasi

DOI: 10.22441/jitkom.2023.v7i1.005

Article History:
Received: Des 21, 2021
Revised: Feb 25, 2023
Accepted: Feb 28, 2023
Published: Mar 29, 2023

I. PENDAHULUAN

Universitas merupakan sebuah perguruan tinggi yang terdiri atas beberapa fakultas yang menyelenggarakan pendidikan ilmiah dalam sejumlah disiplin ilmu tertentu. [1] Adanya universitas tidak dapat dipisahkan dari keberadaan orang-orang yang berperan di dalamnya, salah satunya adalah mahasiswa. Dengan banyaknya mahasiswa yang ada di dalam suatu universitas, pengelolaan data mahasiswa menjadi sangat penting agar dapat terorganisir dengan baik.

Data mahasiswa biasanya disimpan pada server universitas. Selain dibutuhkan kapasitas yang cukup untuk menyimpan data, server juga harus memiliki keamanan yang tinggi. Keamanan dari data tersebut menjadi urgensi yang tidak dapat diabaikan. Data mahasiswa harus dilindungi dari pihak yang tidak berhak mengaksesnya. Keamanan secara umum dinilai berdasarkan tiga faktor utama, yaitu Confidentiality, Integrity, dan Availability atau yang biasa disingkat CIA. Tiga serangkai CIA adalah inti dari keamanan informasi. Pengertian CIA menurut [3] yaitu, Confidentiality berarti bahwa sebuah sistem harus memastikan bahwa hanya pengguna yang berwenang yang dapat mengakses informasi, Integrity berarti bahwa sebuah sistem harus memastikan kelengkapan, keakuratan, dan tidak adanya modifikasi yang tidak sah di semua komponennya, dan Availability berarti bahwa sebuah sistem harus memastikan bahwa semua komponen sistem selalu tersedia dan beroperasi ketika diperlukan oleh pengguna yang berwenang.

Berdasarkan informasi dari [9], ancaman keamanan berupa serangan siber pada tahun 2020 telah terjadi di beberapa universitas, sekolah, dan bahkan rumah sakit. Salah satu kasus serangan siber menimpa University of California, San Francisco (UCSF). Peretas berhasil menyerang server universitas dan mendapatkan data penting berupa data penelitian Fakultas Kedokteran tentang Covid-19. Awalnya peretas meminta tebusan kepada universitas sebanyak US$ 3 juta untuk mendapatkan kunci dekripsi yang dapat mengembalikan data yang telah dienkripsi oleh peretas. Kemudian universitas melakukan negosiasi dengan peretas yang telah mengunci setidaknya tujuh server milik universitas. Universitas terkonfirmasi membayar peretas dengan jumlah

penawaranterakhir dari negosiasi tersebut yaitu sebesar US$ 1,14 juta.

Kasus tersebut membuktikan bahwa data penting yang terletak pada server sebuah organisasi bisa saja diretas dan diakses oleh orang yang tidak berhak. Peretas memiliki tujuan tertentu untuk melakukan penyerangan, seperti mencuri informasi penting, merusak sistem, bersenang-senang, dan juga melakukan pemerasan seperti yang terjadi pada UCSF. Peretas meminta sejumlah uang kepada organisasi, dengan ancaman jika tidak diberi uang tebusan maka data korban yang telah dimiliki oleh peretas tidak akan dikembalikan, atau bahkan sampai disebarluaskan.

Penelitian ini akan mengacu pada beberapa penelitian yang pernah dilakukan berkaitan dengan metode pengujian penetrasi, antara lain:

Addi Amalana Arafat (2020) dalam judul "Penetration Testing pada Website Registrar Pengelola Nama Domain Internet Indonesia (PANDI)", melakukan penelitian untuk mendapatkan informasi tentang kerentanan pada website Registrar PANDI dengan Penetration Testing. Penelitian ini menggunakan metode blackbox dan NIST. Terdapat 10 Daftar Website yang menjadi target pengujian. Pada pengujian ditemukan kerentanan x-header-frame not set yang terbukti dapat merubah tampilan dari 7 website Registrar saat diakses oleh script html. Hasil dari penelitian ini berupa laporan penetration testing yang diberikan kepada pihak Registrar sebagai bahan pertimbangan untuk memperbaiki sistem keamanan.[2]

Rubenson Christian Silaban dan Erick Wijaya (2018) dalam judul "Analisis Kerentanan Website menggunakan Metode NIST SP 800-115 dan OWASP di DISKOMINFO Kabupaten Bandung" melakukan penelitian untuk menganalisis dan menguji keamanan website yang dikelola oleh Diskominfo Kabupaten Bandung dengan menggunakan metode NIST dan OWASP. Hasil pengujian menunjukkan bahwa terdapat kerentanan pada website yang dikelola oleh Diskominfo Kabupaten Bandung berupa SQL Injection dan eksploitasi hak akses sehingga perlu tindakan lebih lanjut untuk menutup celah keamanan tersebut.[4]

Mohd Ehmer dan Farmeena Khan (2012) dalam judul "A Comparative Study of White Box, Black Box and Grey Box Testing Techniques" melakukan penelitian untuk membandingkan tiga teknik pengujian yang ada yaitu White Box, Black Box, dan Grey Box. Ketiga teknik pengujian memiliki kelebihan dan kekurangannya masing-masing. Teknik pengujian White Box dianggap menguntungkan karena cakupan yang maksimum dapat dicapai selama penulisan skenario pengujian. Pada pengujian White Box, penguji juga memiliki pengetahuan yang cukup tentang sistem target.[5]

Girish Janardhanudu dan Ken van Wyk (2013) dalam judul "White Box Testing" bahwa pengujian white box untuk keamanan adalah hal yang berguna dan efektif. Dalam paper ini memperkenalkan bahwa pengujian white box dilakukan berdasarkan pengetahuan tentang bagaimana sistem diimplementasikan serta alat dan teknik yang berlaku untuk pengujian white box untuk keamanan. Pengujian white box dapat dilakukan untuk mengungkap kerentanan yang dapat dieksploitasi. Pengujian white box memerlukan pengetahuan tentang apa yang membuat sistem aman atau tidak aman, cara berpikir seperti penyerang, dan cara menggunakan alat dan teknik pengujian yang berbeda.[6]

Salah satu metode yang digunakan untuk melakukan pengujian penetrasi adalah *National Institute of Standards and Technology* (NIST) yang mengacu pada NIST Special Publication 800-115. Metode ini dirancang untuk membantu organisasi dalam merencanakan dan melakukan pengujian serta memeriksa teknis keamanan informasi, salah satunya dengan melakukan pengujian penetrasi yang terdiri dari 4 fase pengujian, yaitu fase planning, fase discovery, fase attack, dan fase reporting.[7][8] [10]

- Fase Planning

  Fase ini merupakan fase perencanaan tentang pengujian yang akan dilakukan. Beberapa hal yang termasuk dalam perencanaan ini yaitu menentukan target, tujuan, ruang lingkup, serta metode yang digunakan untuk pengujian, serta mendokumentasi persetujuan dalam bentuk Rules of Engagement.

- Fase Discovery

  Fase ini merupakan fase pengumpulan serta pemindaian informasi tentang target pengujian.

- Fase Attack

  Fase ini merupakan fase eksekusi terhadap serangan. Setelah mengeksekusi serangan, kerentanan yang didapat akan diverifikasi dengan cara melakukan exploit terhadap kerentanan tersebut.

- Fase Reporting

  Fase akhir dari pengujian yaitu fase reporting. Fase ini dilakukan bersamaan dengan tiga fase sebelumnya. Saat fase planning dibuat rencana penilaian. Kemudian dalam fase discovery dan attack, log yang tertulis biasanya disimpan dan dibuat laporan secara berkala untuk administrator dan manajemen sistem. Pada akhir pengujian, laporan dibuat untuk mendeskripsikan kerentanan yang teridentifikasi, menyajikan peringkat risiko, dan memberikan rekomendasi tentang cara memitigasi kelemahan yang ditemukan.

Salah satu cara untuk menghindari terjadinya peretasan adalah dengan menutup celah-celah keamanan yang mungkin dimiliki sistem. Sebelum menutup celah keamanan, kita harus mengetahui celah keamanan tersebut dengan melakukan pengujian seperti yang dilakukan oleh peretas, namun dengan prosedur yang telah disetujui. Oleh karena adanya permasalahan tersebut, maka untuk mencegah peretas meretas sistem universitas penulis melakukan penelitian untuk melakukan simulasi serangan terhadap server eksternal sebuah universitas. Penelitian yang dilakukan berjudul Uji Penetrasi Server Universitas PQR Menggunakan Metode *National Institute of Standards and Technology* (NIST SP 800-115).

II. LITERATURE REVIEW

Berdasarkan ilustrasi pada Gambar 1, dalam langkah awal penelitian perlu dilakukan identifikasi dan perumusan masalah untuk menemukan permasalahan yang ada dalam lingkup penelitian. Pada penelitian ini terdapat permasalahan yaitu sistem universitas yang kemungkinan memiliki celah keamanan. Setelah diidentidikasi, ada banyak universitas yang

mengalami kebocoran data karena diserang oleh pihak yang tidak bertanggung jawab. Kebocoran data disebabkan oleh adanya celah keamanan yang tidak terdeteksi pada sistem dan kemudian tidak ditangani dengan baik, sehingga memungkinkan orang lain memasuki sistem tanpa izin. Penulis melakukan penelitian yaitu pengujian penetrasi pada server Universitas PQR menggunakan metode *National Institute of Standards and Technology* (NIST SP 800-115). Penelitian ini bertujuan untuk membantu menemukan celah keamanan yang dapat membahayakan data-data pribadi mahasiswa yang terdapat pada server universitas, sehingga dapat ditangani dengan baik oleh Universitas PQR.

Setelah selesai mengidentifikasi dan merumuskan masalah, selanjutnya penulis melakukan studi literatur. Pada tahap ini, penulis mempelajari penelitian yang telah dilakukan sebelumnya agar penelitian ini dapat berjalan dengan lancar. Penelitian ini berpedoman pada buku, dokumen, serta jurnal yang berkaitan dengan uji penetrasi, keamanan server, dan metode NIST yang menjadi landasan teori serta tahap-tahap pada penelitian. Sumber pustaka pada penelitian ini dicantumkan dalam daftar pustaka.

III. METODOLOGI PENELITIAN

A. *Fase Planning*

Pada fase *planning* penulis melakukan perencanaan tentang pengujian yang akan dilakukan. Beberapa hal yang termasuk dalam perencanaan ini yaitu menentukan target, tujuan, ruang lingkup, serta metode yang digunakan untuk pengujian, melakukan wawancara guna mengetahui fakta di lapangan kepada universitas, melakukan persetujuandengan pihak universitas, mendokumentasi persetujuan dalam bentuk dokumen *Rules of Engagement*.

B. *Fase Discovery*

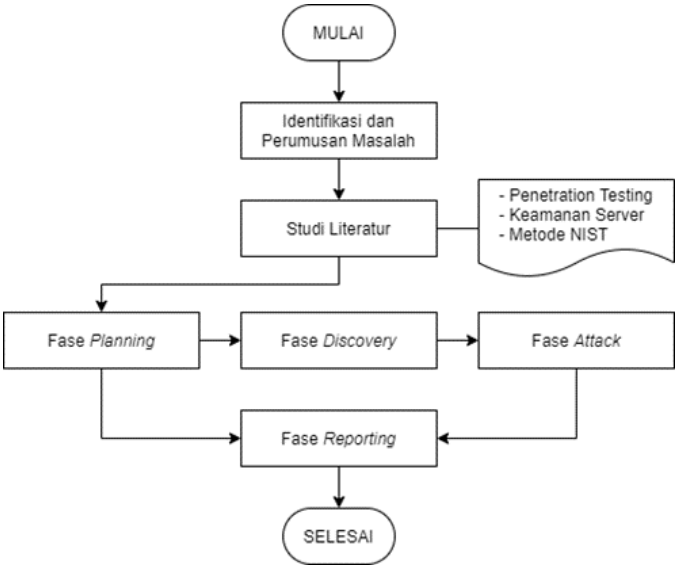
Pada fase *discovery* penulis mengumpulkan serta memindai informasi tentang target pengujian. Pengumpulan serta pemindaian informasi dilaksanakan dengan bantuan *tools* pada Kali Linux 2020.1 yaitu *Network Mapper* (NMAP 7.91). Selain itu juga akan dilakukan analisis kerentanan untuk menemukan kerentanan yang kemungkinan ada pada target pengujian yaitu server Universitas PQR. Analisis kerentanan dilakukan dengan bantuan *tools* Nessus 8.14.0.

C. *Fase Attack*

Pada fase *attack* penulis melakukan eksekusi terhadap serangan. Setelah mengeksekusi serangan, kerentanan yang didapat akan diverifikasi dengan cara melakukan *exploit* terhadap kerentanan tersebut. Jika serangan berhasil dan kerentanan diverifikasi, maka akan diberikan rekomendasi untuk Universitas PQR memitigasi kerentanan yang terdapat pada sistem target. Fase *attack* dilakukan dengan menggunakan berbagai *tools*, yaitu Metasploit 5.0.99, SSL Scan 2.0.0-static, dan Wireshark 3.2.5.

D. *Fase Reporting*

Fase *reporting* dilakukan bersamaan dengan tiga fase sebelumnya. Saat fase *planning* dibuat rencana penilaian. Kemudian dalam fase *discovery* dan *attack*, log yang tertulis biasanya disimpan dan dibuat laporan secara berkala untuk administrator dan manajemen sistem. Pada akhir pengujian, laporan dibuat untuk mendeskripsikan kerentanan yang teridentifikasi, menyajikan peringkat risiko, dan memberikan rekomendasi tentang cara memitigasi kelemahan yang ditemukan.



Gambar 1. Flowchart Tahapan Penelitian

IV. HASIL DAN ANALISA

A. *Fase Planning*

Pada fase *planning*, dibuat perencanaan terhadap pengujian yang akan dilakukan. Perencanaan yang dilakukan yaitu menentukan target pengujian, tujuan pengujian, ruang lingkup pengujian, serta metode yang digunakan untuk pengujian. Perencanaan dibuat berdasarkan hasil diskusi Penguji dan Universitas PQR, setelahnya dilakukan pembuatan peraturan dan persetujuan tentang uji penetrasi yang akan dilakukan, yang didokumentasikan dalam bentuk dokumen *Rules of Engagement.*

B. *Fase Discovery*

Fase discovery dibagi menjadi 2 bagian. Bagian pertama dimulai dengan pengumpulan informasi yang dibutuhkan tentang sistem target untuk pengujian. Pengumpulan informasi dilakukan menggunakan tools dig dan Network Mapper (NMAP) untuk mengumpulkan informasi tentang jaringan uji secara spesifik. Hasil pemindaian dapat dilihat pada tabel berikut.

Tabel 1. Hasil Pemindaian *Port* dengan NMAP

Alamat IP	(Alamat IP Target)			
Nama Domain	(Nama Domain Target)			
Sistem Operasi	Linux 2.6.32 - 3.4 (kemungkinan 91%)			
Port	**Protokol**	**Status**	**Layanan**	**Versi**
22	TCP	Open	SSH	OpenSSH 5.3
80	TCP	Open	HTTP	Apache httpd 2.2.15 (CentOS)
443	TCP	Open	SSL/HTTP	Apache httpd 2.2.15 (CentOS)
5432	TCP	Open	PostgreSQL	PostgreSQL DB 9.6.7 - 9.6.12

Selanjutnya dilakukan *discovery* bagian kedua, yaitu *vulnerability scanning* atau analisis kerentanan. Kegiatan analisis kerentanan dilakukan dengan menggunakan *tools* Nessus dan Nikto. Hasil dari analisis kerentanan dapat dilihat pada tabel 2 dan tabel 3.

Tabel 2. Hasil Analisis Kerentanan Menggunakan Nessus

Severity	CVSS V3.0	Name
CRITICAL	9.8	*PHP < 7.1.33 / 7.2.x < 7.2.24 / 7.3.x < 7.3.11 Remote Code ExecutionVulnerability*
CRITICAL	10.0	*PHP Unsupported Version Detection*
HIGH	7.5	*SSL Version 2 and 3 Protocol Detection*
HIGH	7.5	*DNS Server Spoofed Request Amplification DDoS*
HIGH	7.5	*PHP < 7.3.24 Multiple Vulnerabilities*
HIGH	7.5	*SSL Medium Strength Cipher Suites Supported (SWEET32)*
MEDIUM	6.5	*SSL Certificate Cannot Be Trusted*
MEDIUM	6.4	*SSL Self-Signed Certificate*
MEDIUM	6.5	*TLS Version 1.0 Protocol Detection*
MEDIUM	5.3	*HTTP TRACE / TRACK Methods Allowed*
MEDIUM	5.3	*SSL Certificate Expiry*
MEDIUM	4.3	*SSH Weak Algorithms Supported*
MEDIUM	4.3	*SSL RC4 Cipher Suites Supported (Bar Mitzvah)*

Tabel 3. Hasil Analisis Kerentanan Menggunakan Nikto

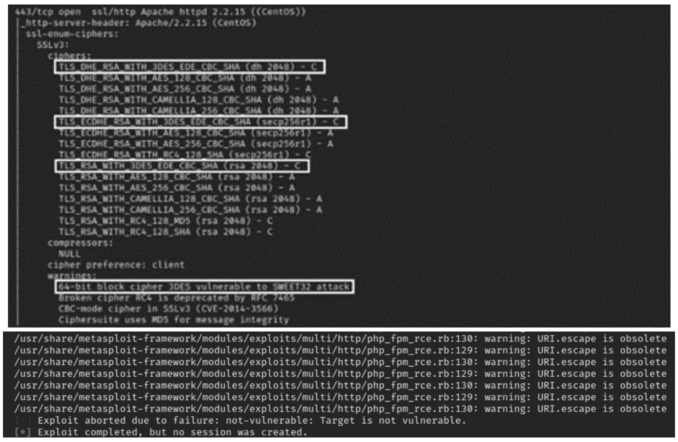
1.	Server: Apache/2.2.15 (CentOS)
2.	Retrieved x-powered-by header: PHP/5.6.40
3.	The anti-clickjacking X-Frame-Options header is not present.
4.	The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
5.	The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
6.	Root page / redirects to: front/gate/index.php
7.	Apache/2.2.15 appears to be outdated (current is at least Apache/2.4.37). Apache 2.2.34 is the EOL for the 2.x branch.
8.	Server may leak inodes via ETags, header found with file /index.html, inode: 32639069, size: 21, mtime: Tue Oct 30 10:30:15 2018
9.	OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable to XST
10.	OSVDB-3268: /includes/: Directory indexing found.
11.	OSVDB-3092: /includes/: This might be interesting...
12.	/info.php: Output from the phpinfo() function was found.
13.	OSVDB-3233: /info.php: PHP is installed, and a test script which runs phpinfo() was found. This gives a lot of system information.
15.	OSVDB-3233: /icons/README: Apache default file found.
16.	OSVDB-5292: /info.php?file=http://cirt.net/rfiinc.txt?: RFI from RSnake's list(http://ha.ckers.org/weird/rfi-locations.dat) or from http://osvdb.org/

C. *Fase Attack*

Pada fase ini dilakukan validasi terhadap kerentanan yang diperoleh dengan cara melakukan eksploitasi kerentanan tersebut. Fase *attack* hanya akan dilakukan pada *port* 80 dan 443. Pengujian pada *port* 22 tidak dilakukan karena universitas tidak memberikan akses terhadap SSH server, lebih baik untuk menutup *port* 22 untuk meminimalisir adanya celah keamanan. Nessus juga tidak mendeteksi adanya kerentanan pada *port* 5432. *Port* 5432 yang menyediakan layanan postgreSQL, memiliki jalur komunikasi yang aman karena telah ada enkripsi yang ditawarkan oleh postgreSQL itu sendiri. Selain itu, untuk peningkatan perlindungan *port* 5432 ini dapat ditutup ketika sedang tidak digunakan. Supaya meminimalkan adanya celah keamanan pada sistem target. Aktivitas validasi kerentanan-kerentanan yang ditemukan akan dijelaskan sebagai berikut.

PHP Remote Code Execution Vulnerability

Kerentanan PHP *Remote Code Execution* ada karena PHP yang berjalan pada sistem merupakan versi lama, yaitu PHP versi 5.6.40. Penyerang jarak jauh dapat melakukan eksploitasi pada kelemahan ini dengan mengirimkan permintaan yang dibuat khusus untuk mengeksekusi sebuah kode tertentu. Peneliti melakukan validasi kerentanan tersebut, namun seperti yang terlihat pada gambar 2 bahwa eksploitasi dibatalkan karena terdapat kegagalan yaitu target tidak rentan. Hal tersebut menunjukkan bahwa kerentanan PHP *Remote Code Execution* yang dianalisis oleh Nessus merupakan *false positive*.



Gambar 2. *Exploit* pada Kerentanan PHP *Remote Code Execution*

PHP Unsupported Version Detection

Kerentanan selanjutnya adalah terdeteksi versi PHP yang sudah tidak didukung. Versi PHP yang digunakan oleh sistem target adalah PHP versi 5.6.40. Pada gambar 3 dapat dilihat bahwa tercantum tanggal akhir masa pakai untuk setiap cabang PHP yang sudah tidak didukung. Versi PHP 5.6.40 pada sistem target merupakan versi PHP yang sudah tidak didukung sejak dua tahun yang lalu.

Gambar 3. Cabang PHP yang Sudah Tidak Didukung Sumber: php.net/eol.php

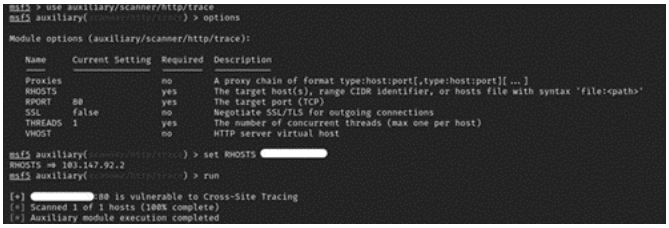
PHP < 7.3.24 Multiple Vulnerability

Sama dengan kerentanan sebelumnya, kerentanan ini ditemukan karena versi PHP yang digunakan pada sistem target merupakan versi PHP sebelum 7.3.24. Versi yang sudah tidak didukung dapat membuat sistem rentan terhadap kerentanan keamanan dan *bug* yang telah diperbaiki di versi PHP yang

lebih baru. Maka, ada baiknya untuk melakukan pembaharuan terhadap versi PHP yang ada pada sistem target agar bisa diperbaiki jika terdeteksi adanya kerentanan serta *bug* pada versi terbaru.

HTTP TRACE / TRACK Methods Allowed

Kerentanan HTTP *Trace* yang diizinkan oleh sistem target memungkinkan untuk mengakses informasi sensitif di *header* HTTP saat membuat permintaan HTTP. Penguji mencoba mengeksploitasi kerentanan tersebut, terlihat pada gambar 4 bahwa sistem target rentan terhadap *Cross Site Tracing*.



Gambar 4. *Exploit* pada kerentanan HTTP *TRACE*

SSL Version 2 and 3 Protocol Detection

Protokol SSL versi 2 dan versi 3 terdeteksi aktif, Peneliti menggunakan *tools* SSLScan untuk melakukan pemindaian dan validasi terhadap kerentanan tersebut, apakah sistem target sudah menggunakan keamanan SSL atau belum. Hasil pemindaian pada gambar 5 menunjukkan bahwa SSL versi 2 *disabled* dan SSL versi 3 *enabled*. SSL versi 3 dianggap memiliki keamanan yang rendah sehingga dapat dimanfaatkan oleh penyerang untuk masuk ke dalam sistem. Maka direkomendasikan untuk segera menonaktifkan protokol SSL versi 3.

Gambar 5. Pemindaian SSL

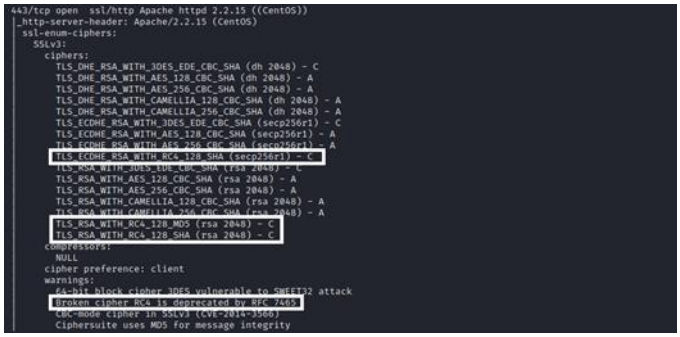
SSL Medium Strength Cipher Suites Supported (SWEET32)

Pengujian terhadap kerentanan SSL mendukung *cipher* berkekuatan sedang, dilakukan dengan pemindaian NMAP. Hasil pemindaian pada gambar 6 menunjukkan daftar *cipher* berkekuatan sedang yang didukung oleh sistem target. *Cipher* tersebut memiliki nilai huruf C yang berarti kekuatan *cipher*nya adalah sedang dan berada di bawah nilai huruf A dan B. Hasil pemindaian juga memberikan peringatan bahwa blok *cipher* 64-bit 3DES rentan terhadap serangan SWEET32.

Gambar 6. *Pemindaian ssl-enum-ciphers*

TLS Version 1.0 Protocol Detection

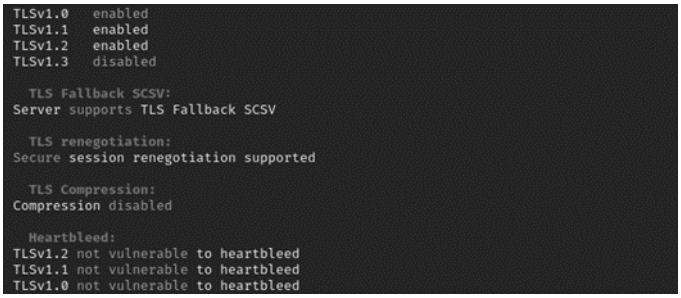
Nessus mendeteksi protokol TLS versi 1.0, yang merupakan TLS versi terdahulu. TLS versi 1.0 memiliki sejumlah kelemahan kriptografi. Peneliti mencoba melakukan validasi kerentanan menggunakan *tools* SSLScan. Hasil yang didapatkan dapat dilihat pada gambar 7 terlihat bahwa TLS versi 1.0, versi 1.1, versi 1.2 berstatus diaktifkan, sedangkan TLS versi 1.3 dinonaktifkan.



Gambar 7. Pemindaian TLS

SSL Certificate

Terdeteksi 3 kerentanan tentang sertifikat SSL, yaitu *SSL Certificate Cannot Be Trusted, SSL Self- Signed Certificate,* dan *SSL Certificate Expiry*. Hasil pengujian pada gambar 8 menunjukkan bahwa sertifikat SSL sistem target telah kadaluarsa. Sertifikat tidak valid setelah 30 Oktober 2019. Hal itu menunjukkan bahwa layanan SSL pada sistem target sudah berakhir dan sertifikat SSL menjadi tidak dapat dipercaya. Dengan begitu dapat memudahkan penyerang untuk menyusup ke dalam sistem, karena sistem tidak lagi memiliki SSL. Gambar tersebut juga menjelaskan bahwa sertifikat tidak mengandung ekstensi *Issuers Certificate Authority*, serta adanyakecocokan pada *Subject* dan *Issuer* yang berarti memungkinkan adanya sertifikat SSL *Self-Signed*.



Gambar 8. Validasi Kerentanan Sertifikat SSL

SSL RC4 Cipher Suites Supported (Bar Mitzvah)

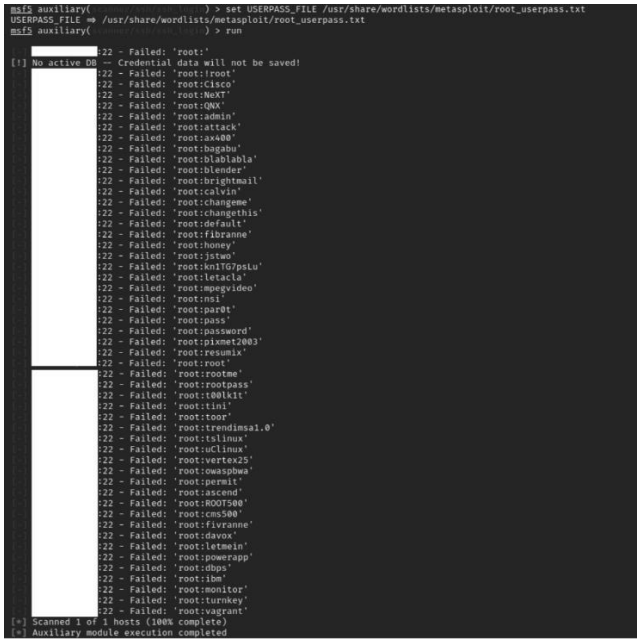
SSL mendukung penggunaan *cipher* RC4 dalam satu atau lebih rangkaian *password*. *Cipher* RC4 dianggap memiliki kelemahan dalam generasi aliran *pseudo-random byte*. Jika plainteks berulang kali dienkripsi dan penyerang dapat memperoleh banyak cipherteks, maka penyerang juga dapat memperoleh plainteks. Validasi kerentanan dilakukan dengan bantuan *tool* NMAP. Dari gambar 9 dapat dilihat bahwa terdapat beberapa *cipher* RC4 yang didukung oleh sistem target. *Cipher* tersebut memiliki nilai huruf C yang berarti kekuatan *cipher*nya adalah sedang. Hasil pemindaian juga memberikan peringatan bahwa *cipher* RC4 yang rusak sudah tidak digunakan lagi oleh RFC 7465.

Gambar 9. Pemindaian *ssl-enum-ciphers*

SSH Brute Force

Peneliti mencoba melakukan password cracking dengan brute force attack pada sistem target. Peneliti menggunakan wordlists atau file yang berisi kumpulan kata dari Metasploit yaitu root_userpass untuk melakukan SSH Login.

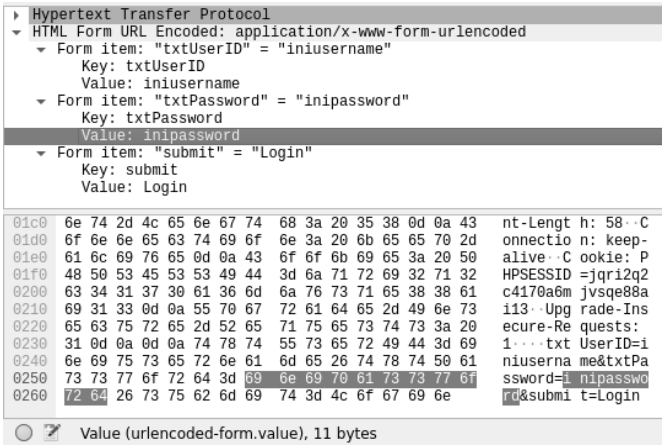
Dapat dilihat pada gambar 10 bahwa hasil percobaan *password cracking* dengan menggunakan *wordlists* yang dilakukan oleh peneliti tidak dapat menembus sistem target. Namun sistem target termasuk rentan terhadap *brute force attack* karena tidak menerapkan pembatasan upaya untuk *login*. Supaya sistem lebih aman, seharusnya dilakukan pembatasan upaya percobaan *login* pada SSH dengan jumlah gagal paling banyak yaitu 5 kali. Denganmembuat pembatasan upaya *login* yang gagal, akan mengunci *user* jika mereka memasukkan *password* yang salah lebih dari jumlah yang telah ditentukan.



Gambar 10. *SSH Brute Force*

Unencrypted Password Form

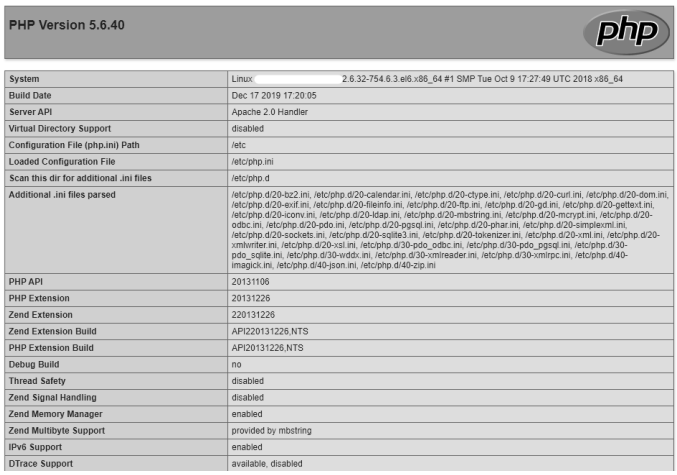
Peneliti mencoba melakukan penyadapan pada jaringan target dengan menggunakan Wireshark, yang dapat mengidentifikasi paket-paket yang berjalan pada jaringan. Dari gambar 11 dapat dilihat bahwa pada *HTML Form URL*, berisi *username* dan *password* yang diinput tidak terenkripsi. Penyadapan berhasil dilakukan karena dapat dilihat dengan jelas plainteks dari data tersebut. Inputan yang tidak terenkripsi disebabkan oleh penggunaan *HTTP* daripada *HTTPS*. Jika terdapat layanan *HTTPS*, lebih baik menonaktifkan layanan *HTTP* atau port 80. Sehingga yang digunakan hanyalah *Secure HTTP*.



Gambar 11. *Network Sniffing* dengan Wireshark

PHPinfo() Information Disclosure

Peneliti mencoba melakukan pemindaian http files_dir pada Metasploit. Dari hasil pemindaian ditemukan beberapa file yang menarik pada target. Beberapa diantaranya merupakan informasi yang sama seperti yang ditemukan pemindai kerentanan Nikto. Peneliti mencoba untuk mengakses salah satu file yaitu http://(alamat IP target):80/info.php yang menunjukkan adanya informasi tentang PHP yang dapat diakses oleh publik. Informasi yang disediakan adalah seperti yang terlihat pada gambar 12. Informasi yang dibocorkan oleh /info.php mencakup banyak hal, contohnya seperti informasi tentang jalur fisik, variabel lingkungan, dan pengaturan konfigurasi PHP yang lengkap. Informasi tersebut tentunya berbahaya jika terus disediakan untuk publik.

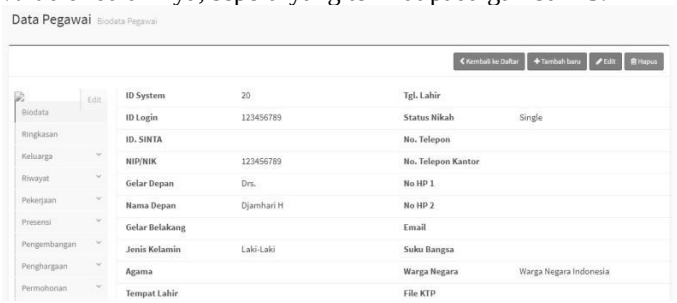


Gambar 12. Informasi yang ada pada /info.php

Default Credential

Peneliti mencoba melakukan *password cracking* dengan teknik *guessing username* dan *password* di *form login* yang terdapat pada halaman awal server. Setelah beberapa kali percobaan, ditemukan satu kombinasi *username* dan *password* yang sesuai sehingga Peneliti dapat login ke dalam sistem. Setelah berhasil masuk, peneliti mencoba mencari informasi penting di dalam sistem. Akun yang berhasil disusupi oleh peneliti ternyata merupakan akun dosen yang tidak terpakai,

terbukti dengan banyaknya data pribadi yang tidak terdapat *value* di dalamnya, seperti yang terlihat pada gambar 13.

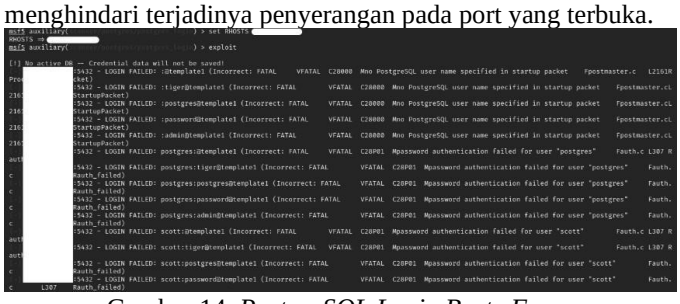


Gambar 13. Data Pribadi Pegawai dalam Sistem Target

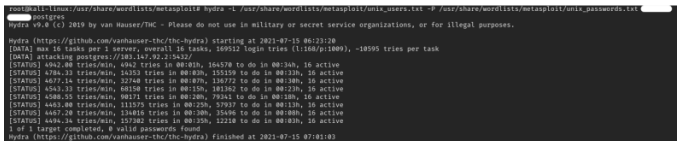
Meskipun tidak adanya informasi penting yang bisa diakses, namun kombinasi *username* dan *password* yang didapatkan merupakan sebuah kerentanan keamanan yaitu *default credentia*l. *Username* dan *password* yang ada seharusnya diubah menjadi rumit seperti menggunakan minimal 12 karakter atau lebih. *Password* dapat berupa kombinasi dari huruf, angka, dan simbol. Maka direkomendasikan untuk menghapus akun yang tidak dipakai, atau mengganti *username* dan *password default* menjadi kombinasi yang lebih rumit.

PostgreSQL Login Brute Force

Peneliti mencoba melakukan *password cracking* dengan *brute force attack* pada *login postgreSQL* yang terletak di *port* 5432. *Password cracking* dilakukan dengan menggunakan salah satu *auxiliary* pada Metasploit yaitu *Postgres Login* dan juga menggunakan *wordlists* pada *hydra.* Hasil pengujian dapat dilihat pada gambar 14 dan 15. *Password cracking* pada *postgres login* yang dilakukan oleh peneliti tidak dapat menembus sistem target. Namun sistem target termasuk rentan terhadap *brute force attack* karena tidak menerapkan pembatasan upaya untuk *login*. Supaya sistem lebih aman, seharusnya dilakukan pembatasan upaya percobaan *login* pada SSH dengan jumlah gagal paling banyak yaitu 5 kali. Dengan membuat pembatasan upaya *login* yang gagal, akan mengunci *user* jika mereka memasukkan *password* yang salah lebih dari jumlah yang telah ditentukan. Jika memang port layanan postgreSQL tidak digunakan, lebih baik untuk ditutup saja guna menghindari terjadinya penyerangan pada port yang terbuka.



Gambar 14. *PostgreSQL Login Brute Force*



Gambar 15. *PostgreSQL Brute Force* dengan *Hydra*

D. Fase Reporting

Fase *reporting* dikembangkan untuk mendeskripsikan kerentanan yang ditemukan, tingkat risiko, dan memberikan petunjuk bagaimana cara untuk memitigasi kelemahan yang ditemukan. Fase *reporting* terjadi bersamaan dengan tiga fase lainnya, yaitu saat berjalannya fase *planning*, *discovery*, dan *attack*. Kerentanan yang ditemukan Peneliti pada pengujian penetrasi di penelitian ini adalah 13 kerentanan yang dapat dieksploitasi dengan rincian 2 kerentanan termasuk kategori *critical*, 3 kerentanan termasuk kategori *high*, 8 kerentanan termasuk kategori *medium* dan 1 kerentanan adalah *false positive*. Kerentanan-kerentanan yang ditemukan tentunya perlu ditindaklanjuti oleh organisasi terkait, supaya dapat dilakukan perbaikan.

Tabel 4. Hasil Uji Penetrasi

No.	Skor Kerentanan		Nama Kerentanan	Hasil Pengujian	Rekomendasi Mitigasi
	Severity	CVSS V3.0			
1	***CRITICAL***	**10.0**	*Default Credential*	Rentan ditemukannya kombinasi *username* dan *password* yang cocok untuk masuk ke sistem	*Username* dan *password* yang ada seharusnya diubah menjadi rumit seperti melakukan kombinasi dari huruf kapital, huruf biasa, angka, dan *symbol* serta mengatur banyaknya *character* yang lebih dari 8.
2	***CRITICAL***	**9.8**	*PHP < 7.1.33 / 7.2.x < 7.2.24 / 7.3.x < 7.3.11 Remote Code Execution Vulnerability.*	Tidak rentan	-
3	***CRITICAL***	**10.0**	*PHP Version Unsupported Detection*	Rentan, versi PHP yang digunakan merupakan versi using yaitu PHP 5.6.40	Melakukan pembaharuan pada PHP menjadi versi PHP terbaru yang didukung yang didukung
4	***HIGH***	**7.5**	*SSL Version 2 and 3 Protocol Detection*	Rentan, protokol SSL 2 dan 3 terdeteksi aktif	Menonaktifkan protokol SSL
5	***HIGH***	**7.5**	*PHP < 7.3.24 PHP < 7.3.24 PHP < 7.3.24 Multiple Vulnerabilities*	Rentan, versi PHP digunakan merupakan usang yaitu PHP 5.6.40	Melakukan pembaharuan pada PHP menjadi versi PHP terbaru yang didukung
6	***HIGH***	**7.5**	*SSL Medium Strength Cipher Suites Supported (SWEET32)*	Rentan, terbukti didukungnya Rentan terbukti didukungnya *cipher* SSLberkekuatan sedang	Melakukan konfigurasi ulang terhadap aplikasi yang terpengaruh serta menghindari penggunaan cipher

					berkekuatan sedang
7	***MEDIUM***	**6.5**	*SSL Certificate Cannot Be Trusted*	Rentan, sertifikat SSL sudah	Melakukan pembaharuan terhadap sertifikat SSL
8	***MEDIUM***	**6.4**	*SSL Self-Signed Certificate*		Melakukan pembaharuan terhadap sertifikat SSL
9	***MEDIUM***	**6.5**	*TLS Version 1. 0 Protocol Detection*		Menonaktifkan protokol TLS versi 1.0 dan versi 1.1 Mengaktifkan protokol TLS versi 1.2 dan versi 1.3
10	***MEDIUM***	**5.8**	*PHPinfo() Information Disclosure*		Menghapus file info.php pada sistem
11	***MEDIUM***	**5.3**	*Unencrypted PasswordForm*		Menggunakan protokol enkripsi yang terbaru dan paling aman, yaitu TLS versi 1.2 dan versi 1.3 Protokol versi lama seperti SSL versi 1, SSL versi 2, TLS versi 1.0, TLS versi 1.1, dan sandi lemah (< 128 bit) juga harus dinonaktifkan
12	***MEDIUM***	**5.3**	*HTTP TRACE / TRACK Methods Allowed*		Menonaktifkan metode Trace pada HTTP
13	***MEDIUM***	**5.3**	*SSL Certificate Expiry*		Melakukan pembaharuan terhadap sertifikat SSL
14	***MEDIUM***	**4.3**	*SSL RC4 Cipher Suites Supported (Bar Mitzvah)*		Melakukan konfigurasi ulang terhadap aplikasi yang terpengaruh serta menghindari penggunaan cipher RC4

Sebaiknya pihak Universitas juga perlu menutup *port* layanan yang tidak terpakai, untuk meminimalkan pintu masuk serangan dan risiko terjadinya penyerangan terhadap *port* yang berstatus terbuka. Selain rekomendasi teknis yang diberikan, juga terdapat beberapa rekomendasi non-teknis untuk pihak Universitas PQR. Rekomendasi non-teknis adalah sebagai berikut.

1. Melakukan perbaikan terhadap sistem adalah satu hal penting yang harus segera dilakukan. Jika tidak segera dilakukan perbaikan, tentu dapat memberikan dampak yang buruk bagi kelangsungan bisnis Universitas seperti kehilangan data-data yang ada pada server eksternal Universitas PQR.
2. Setelah dilakukan perbaikan, sebaiknya dilakukan pengujian kembali terhadap sistem. Uji penetrasi sebaiknya rutin dilakukan setidaknya 2 (dua) kali dalam satu tahun. Dengan adanya pengujian penetrasi yang rutin, dapat dilakukan pendokumencegahan serta perbaikan terhadap keamanan sistem Universitas PQR, sehingga menjadi lebih aman dari waktu ke waktu.
3. Membuat dokumentasi serta kumpulan aturan keamanan dari sistem organisasi, untuk memudahkan pengelolaan data serta kontrol keamanan pada server eksternal Universitas PQR.

Kerentanan-kerentanan yang ditemukan dapat dimitigasi dengan cara yang berbeda-beda untuk setiap kerentanan. Peneliti hanya memberikan rekomendasi yang dapat dilakukan untuk memitigasi kerentanan yang ditemukan. Penerapan dari rekomendasi akan diserahkan sepenuhnya kepada pihak terkait, yaitu Universitas PQR.

V. KESIMPULAN

Setelah melakukan pengujian penetrasi pada server Universitas PQR, dapat disimpulkan beberapa hal berikut.

1. Pengujian penetrasi terhadap keamanan data pribadi mahasiswa pada server Universitas PQR dilakukan dengan melalui 4 fase pengujian, yaitu fase *planning*, fase *discovery*, fase *attack*, dan fase *reporting*.
2. Kerentanan yang ditemukan sebagai hasil dari pengujian penetrasi adalah ditemukannya 13 kerentanan yang dapat dieksploitasi dengan rincian 2 kerentanan termasuk kategori *critical yaitu Default Credentials* dan *PHP Unsupported Version Detection*, 3 kerentanan termasuk kategori *high yaitu SSL Version 2 and 3 Protocol Detection*, *PHP < 7.3.24 Multiple Vulnerabilities, SSL Medium Strength Cipher Suites Supported (SWEET32)*, 8 kerentanan termasuk kategori *medium* yaitu *SSL Certificate Cannot Be Trusted, SSL Self-Signed Certificate, TLS Version 1.0 Protocol Detection, PHPinfo() Information Disclosure, Unencrypted Password Form, HTTP TRACE / TRACK Methods Allowed, SSL Certificate Expiry, SSL RC4 Cipher Suites Supported (Bar Mitzvah)*, dan 1 kerentanan adalah *false positive yaitu PHP < 7.1.33 / 7.2.x < 7.2.24 / 7.3.x < 7.3.11 Remote Code Execution Vulnerability*.
3. Kerentanan-kerentanan yang ditemukan dapat dimitigasi dengan cara yang berbeda-beda untuk setiap kerentanan. Rekomendasi untuk kerentanan *Default Credential* yaitu *username* dan *password* yang ada seharusnya diubah menjadi rumit seperti melakukan kombinasi dari huruf kapital, huruf biasa, angka, dan *symbol* serta mengatur banyaknya *character* yang lebih dari 8. Rekomendasi untuk kerentanan *PHP Unsupported Version Detection* dan *PHP < 7.3.24 Multiple Vulnerabilities* yaitu melakukan pembaharuan pada PHP menjadi versi PHP terbaru yang didukung. Rekomendasi untuk kerentanan *SSLVersion 2 and 3 Protocol Detection* yaitu

menonaktifkan protokol SSL. Rekomendasi untuk kerentanan *SSL Medium Strength Cipher Suites Supported (SWEET32)* yaitu melakukan konfigurasi ulang terhadapaplikasi yang terpengaruh serta menghindari penggunaan cipher berkekuatan sedang. Rekomendasi untuk kerentanan *SSL Certificate Cannot Be Trusted, SSL Self-Signed Certificate*, serta *SSL Certificate Expiry* yaitu melakukan pembaharuan terhadap sertifikat SSL. Rekomendasi untuk kerentanan *TLS Version 1.0 Protocol Detection* yaitu menonaktifkan protokol TLS versi 1.0 dan versi 1.1 dan mengaktifkan protokol TLS versi 1.2 dan versi 1.3. Rekomendasi untuk kerentanan *PHP info() Information Disclosure yaitu menghapus file info.php pada sistem. Rekomendasi untuk kerentanan Unencrypted Password Form yaitu enggunakan protokol enkripsi yang terbaru dan paling aman, yaitu TLS versi 1.2 dan versi 1.3 Protokol versi lama seperti SSL versi 1, SSL versi 2, TLS versi 1.0, TLS versi 1.1, dan sandi lemah (< 128 bit) juga harus dinonaktifkan Rekomendasi untuk kerentanan HTTP TRACE / TRACK Methods Allowed yaitu menonaktifkan metode Trace pada HTTP. Rekomendasi untuk kerentanan SSL RC4 Cipher Suites Supported (Bar Mitzvah) yaitu melakukan konfigurasi ulang terhadap aplikasi yang terpengaruh serta menghindari penggunaan cipher RC4. Peneliti hanya memberikan rekomendasi yang dapat dilakukan untuk memitigasi kerentanan yang ditemukan. Penerapan dari rekomendasi akan diserahkan sepenuhnya kepada pihak terkait, yaitu Universitas PQR.*

4. Hasil pengujian menunjukkan bahwa server universitas PQR masih rentan, sehingga diperlukan penanganan serta perbaikan kerentanan tersebut oleh pihak Universitas PQR.

Berdasarkan pada pembahasan dan kesimpulan yang telah dijelaskan, maka Penulis memberikan beberapa saran yaitu sebagai berikut.

1. Pengujian selanjutnya dapat dilakukan dengan menambahkan metode threat hunting pada pengujan supaya dapat menemukan kerentanan secara lebih mendalam.
2. Pengujian ulang pada sistem sebaiknya dilakukan kembali jika sistem telah berhasil diremediasi oleh Universitas PQR.

DAFTAR PUSTAKA

[1] Kamus Besar Bahasa Indonesia. [Online]. Diambil pada tanggal 24 Oktober 2020 dari https://kbbi.kemdikbud.go.id/entri/universitas

[2] A. A. Arafat, "Penetration testing pada website registrar Pengelola Nama Domain Internet Indonesia (PANDI)," *Uinjkt.ac.id*, 2020, doi: http://repository.uinjkt.ac.id/dspace/handle/123456789/53637.

[3] Y. Cherdantseva and J. Hilton, "A Reference Model of Information Assurance & Security," 2013 International Conference on Availability, Reliability and Security, Regensburg, Germany, 2013, pp. 546-555, doi: 10.1109/ARES.2013.72.

[4] C. S. Rubenson, "Analisis Kerentanan Website Menggunakan Metode Nist Sp 800-115 Dan Owasp Di Diskominfo Kabupaten Bandung - Repository," Unikom.ac.id, Nov. 2018.

[5] A. Kumar, K. Vinod, and C. Narottam. "Energy efficient clustering and cluster head rotation scheme for wireless sensor networks." *International Journal of Advanced Computer Science and Applications*, vol. 3, no. 5 2012.

[6] G. Janardhanudu, and K. Wyk, "*White-box Testing*." 2013. [Online]. Diambil pada tanggal 28 Juni 2021 dari https://us-cert.cisa.gov/bsi/articles/best-practices/white-box-testing/white-box-testing

[7] K. Scarfone, M. Souppaya, A. Cody, and A. Orebaugh, "*Special Publication 800-115 Technical Guide to Information Security Testing and Assessment Recommendations of the National Institute of Standards and Technology."* 2008. *[Online]. Diambil pada tanggal 24 Oktober 2020 dari https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-115.pdf*

[8] R. A. Wibowo and S. Widyarto, "Kajian Pustaka: Penetration Testing Dengan Nist SP 800-115 Dan OSSTMM," *Proceedings of the Informatics Conference*, vol. 6, no. 10, pp. 96–111, 2020, Accessed: Apr. 09, 2023. [Online]. Available: https://ojs.journals.unisel.edu.my/index.php/icf/article/view/96

[9] J. Tidy, "How hackers extorted $1.14m from a US university." *BBC News*, 2020. Diambil pada tanggal 3 Desember 2020 dari https://www.bbc.com/news/technology-53214783

[10] W. Wardana, A. Almaarif, and A. Widjajarto, "Vulnerability assessment and penetration testing on the xyz website using NIST 800-115 standard." *J. Ilm. Indones*, vol. 7, no. 1, pp. 520-529, 2022.