Procedia of Engineering and Life Science Vol. 4 June 2023
Seminar Nasional & Call Paper Fakultas Sains dan Teknologi (SENASAINS 6th)
Universitas Muhammadiyah Sidoarjo

Detection and Prevention of Insecure Direct Object References (IDOR) in Website-Based Applications

Deteksi dan Pencegahan Insecure Direct Object References (IDOR) Pada Aplikasi Berbasis Website

Rio Ananda Putra[1], Irwan Alnaurus Kautsar[2], Hindarto[3], Sumarno[4]
{191080200186@umsida.ac.id[1], irwan@umsida.ac.id[2], hindarto@umsida.ac.id[3], sumarno@umsida.ac.id[4]}

Program Studi Informatika, Fakultas Sains dan Teknologi, Universitas Muhammadiyah Sidoarjo

Abstract. *IDOR (Insecure Direct Object References) is a security vulnerability that occurs when a web application does not validate or authorize access to direct objects, such as data or resources, in an adequate manner. In the context of web application security, objects can be files, database records, or other resources identified by a parameter or direct reference. The IDOR technique allows an attacker to manipulate parameters passed to a web application to gain unauthorized access to objects he or she should not have access to. By exploiting this vulnerability, attackers can access, modify, or delete data that should only be accessible to authorized users. One of the dangers in accessing data on websites, data retrieval techniques from object IDs are often vulnerable to Insecure Direct Object References (IDOR) attacks. Therefore, the data retrieval technique from $_SESSION can be a safer alternative to avoid the IDOR security vulnerability. Using this technique, only the account in use can be accessed and does not allow access to other technician accounts. The use of additional query parameters can also increase website security and protect the data and information contained therein. Thus, adding additional validation to the code can help prevent IDOR vulnerabilities from occurring in web applications.*
Keywords *— Insecure Direct Object References (IDOR); Website; Website Security*

Abstrak. *IDOR (Insecure Direct Object References) adalah kerentanan keamanan yang terjadi saat aplikasi web tidak memvalidasi atau mengotorisasi akses terhadap objek langsung, seperti data atau sumber daya, dengan cara yang memadai. Dalam konteks keamanan aplikasi web, objek bisa berupa file, rekaman database, atau sumber daya lainnya yang diidentifikasi oleh sebuah parameter atau referensi langsung. Teknik IDOR memungkinkan penyerang untuk memanipulasi parameter yang diteruskan ke aplikasi web untuk mendapatkan akses tidak sah ke objek yang seharusnya tidak dapat diaksesnya. Dengan memanfaatkan kerentanan ini, penyerang dapat mengakses, memodifikasi, atau menghapus data yang seharusnya hanya dapat diakses oleh pengguna yang diotorisasi. Salah satu bahaya dalam mengakses data pada website, teknik pemanggilan data dari objek ID seringkali rentan terhadap serangan Insecure Direct Object References (IDOR). Oleh karena itu, teknik pemanggilan data dari $_SESSION dapat menjadi alternatif yang lebih aman untuk menghindari kerentanan keamanan IDOR. Dengan menggunakan teknik ini, hanya akun yang digunakan yang dapat diakses dan tidak memungkinkan akses ke akun teknisi lainnya. Penggunaan parameter query tambahan juga dapat meningkatkan keamanan website dan melindungi data serta informasi yang terdapat di dalamnya. Dengan demikian, penambahan validasi tambahan pada kode tersebut dapat membantu mencegah terjadinya kerentanan IDOR pada aplikasi web.*
Kata kunci — *Insecure Direct Object References (IDOR); Website; Keamanan website*

I. PENDAHULUAN

Teknologi informasi merupakan bidang yang berkembang sangat pesat dalam beberapa tahun terakhir. Hal ini terlihat dari banyaknya inovasi dan penemuan baru yang terus muncul dan merubah cara manusia berinteraksi dan berkomunikasi. Salah satu bentuk penggunaan teknologi informasi yang semakin populer adalah website. Website telah menjadi salah satu media informasi yang sangat penting dan banyak digunakan oleh masyarakat [1].

Teknologi informasi web yang semakin maju pastinya diikuti juga dengan kerentanan keamanan website. Kerentanan keamanan website adalah celah yang memungkinkan penyerang untuk mengakses atau mengubah informasi yang disimpan di sebuah website. Kerentanan ini dapat menyebabkan kerugian data, pencurian identitas, dan bahkan kerusakan sistem. Kerentanan keamanan website dapat disebabkan oleh berbagai faktor, termasuk kode yang buruk, dan konfigurasi yang tidak aman [2].

Insecure Direct Object References (IDOR) adalah sebuah kerentanan keamanan yang disebkan karenanya lemahnya autorisasi pada suatu sistem. Kerentanan ini muncul ketika aplikasi menggunakan input user untuk

mengakses objek pada saat sistem secara langsung, maka hal tersebut memungkinkan user untuk merubah data dan mengakases user lain [3].

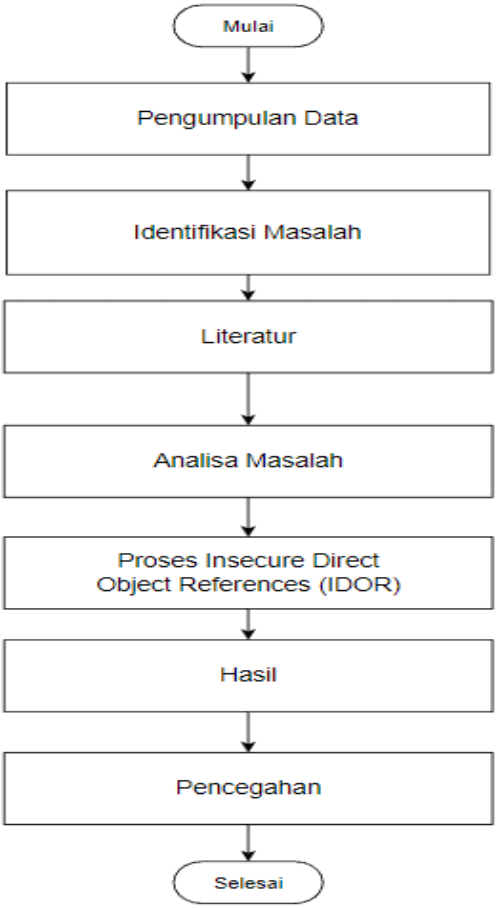
IDOR memiliki peran utama dalam pelanggaran dan skandal keamanan besar di seluruh dunia. Berikut ini contoh kasus kerentanan IDOR (*Insecure Direct Object References*). Peneliti Keamanan AS menemukan kasus IDOR mengenai Samsung, lebih dari 1,5 Juta catatan dari database yahoo telah dihapus oleh satu orang Mesir menggunakan IDOR, peneliti keamanan Turki meretas situs Pengembang Apple menggunakan IDOR. [4] dan pada tahun 2022 kemarin peneliti menemukan bug IDOR pada website simanta umsida

Berdasarkan fakta di atas maka menjadi latar belakang penelitian ini. Peneliti membuat web uji coba sebagai bahan penelitian. Penelitian ini bertujuan menghindari atau menanggulangi adanya kerentanan keamanan IDOR (*Insecure Direct Object References*).

II. METODE

Terdapat beberapa tahapan penelitian pengumpulan data yang digunakan pada penelitian ini sebagai berikut:

A. Teknik analisa



Gambar 1 Teknik Analisa

B. Pengumpulan data

Tahapan pengumpulan data dimulai dengan melakukan analisa terlebih dahulu terhadap website yang menjadi target. Data yang akan digunakan dalam analisa yaitu mendata menu web, mendata level akses web, dan mendata setiap kategori data.

C. Identifikasi masalah

Setelah tahapan pengumpulan data maka proses identifikasi dilakukan guna merumuskan masalah yang ada pada website untuk nantinya dilakukan pengecekan Insecure Direct Object References (IDOR). Berikut langkah identifikasi masalah:

1. Pemahaman Fungsionalitas Aplikasi: Langkah pertama adalah memahami fungsionalitas aplikasi secara menyeluruh. Perlu meneliti bagaimana objek atau data sensitif diidentifikasi dan diakses dalam aplikasi. Perhatikan bagaimana aplikasi membedakan dan mengontrol akses pengguna terhadap objek-objek ini.
2. Identifikasi Objek dan Parameter: Selanjutnya, identifikasi objek atau parameter yang berpotensi rentan terhadap serangan IDOR. Cari tahu bagaimana aplikasi mengenal objek-objek ini dan apakah ada mekanisme pengendalian akses yang lemah atau tidak memadai.
3. Eksplorasi Aplikasi: Coba akses objek-objek yang berpotensi sensitif atau terbatas pada akun pengguna tertentu. Periksa apakah ada kemungkinan untuk memanipulasi parameter atau data yang dikirimkan ke server untuk mendapatkan akses yang tidak sah ke objek tersebut. Perhatikan apakah ada perbedaan antara objek yang seharusnya tidak dapat diakses oleh pengguna dan objek yang dapat diakses dengan menggunakan teknik IDOR.
4. Verifikasi Kekurangan Keamanan: Jika berhasil mengakses objek yang tidak seharusnya diakses, verifikasi apakah akses tersebut tidak sah atau tidak terotorisasi. Pastikan bahwa kerentanan IDOR benar-benar ada dan bahwa akses yang diperoleh melalui manipulasi parameter atau data adalah pelanggaran terhadap mekanisme kontrol akses yang seharusnya ada.
5. Pelaporan dan Rekomendasi: Setelah berhasil mengidentifikasi kerentanan IDOR, dokumentasikan temuan secara rinci dan laporkan kepada pemilik aplikasi atau tim pengembang yang bertanggung jawab. Berikan rekomendasi yang jelas dan konstruktif tentang bagaimana masalah dapat diperbaiki dan keamanan dapat ditingkatkan.

D. Literatur

Pada penelitian yang kami lakukan menggunakan metode Insecure Direct Object References (IDOR). Metode tersebut digunakan berdasarkan acuan dari beberapa referensi dari berbagai sumber diantaranya:

1. Jurnal Keamanan Informasi: Cari jurnal yang mengkhususkan diri dalam keamanan informasi atau keamanan aplikasi. Jurnal-jurnal ini sering menyajikan penelitian terbaru tentang kerentanan dan teknik serangan, termasuk IDOR. Contohnya, Journal of Computer Security dan ACM Transactions on Information and System Security.
2. Konferensi Keamanan: Konferensi-konferensi keamanan komputer dan aplikasi juga merupakan sumber yang baik untuk merujuk. Contohnya, Conference on Computer and Communications Security (CCS), Black Hat Conference, dan DEF CON. Artikel dan presentasi dari konferensi-konferensi ini seringkali mencakup penelitian terkait IDOR dan kerentanan lainnya.
3. Buku dan Pedoman Praktik Terkait Keamanan Aplikasi: Buku-buku dan pedoman praktik tentang pengujian keamanan aplikasi juga dapat memberikan wawasan yang berguna tentang IDOR dan teknik serangan lainnya. Beberapa contoh buku yang populer adalah "The Web Application Hacker's Handbook" oleh Dafydd Stuttard dan Marcus Pinto, serta "OWASP Testing Guide" yang diterbitkan oleh Open Web Application Security Project (OWASP).
4. Publikasi OWASP: OWASP adalah organisasi nirlaba yang fokus pada keamanan aplikasi web. Mereka menyediakan banyak materi yang bermanfaat tentang kerentanan dan teknik serangan, termasuk IDOR. Lalu dapat mencari dokumentasi dan referensi terkait IDOR di situs web mereka dan mengacu pada panduan dan materi referensi yang mereka sediakan.

E. Analisa masalah

Pada tahapan ini peneliti melakukan proses analisa masalah dalam perumusan solusi guna menyelesaikan masalah. Tahapan ini peneliti melakukan tahapan analisa yaitu:

1. Identifikasi Sistem atau Aplikasi yang Akan Dianalisis: Tentukan sistem atau aplikasi yang akan menjadi fokus penelitian. Pilihlah sistem yang mengandung elemen yang memungkinkan terjadinya kerentanan IDOR, seperti akses ke objek yang terkait dengan pengguna.
2. Pemahaman Fungsionalitas Aplikasi: Teliti secara menyeluruh fungsionalitas aplikasi yang akan dianalisis. Pahami bagaimana objek-objek diidentifikasi, diakses, dan terkait dengan pengguna. Perhatikan bagaimana mekanisme kontrol akses diterapkan untuk mengatur hak akses pengguna terhadap objek-objek tersebut.
3. dentifikasi Objek yang Rentan: Identifikasi objek-objek dalam aplikasi yang berpotensi rentan terhadap serangan IDOR. Fokus pada objek-objek yang seharusnya tidak dapat diakses oleh pengguna tertentu, tetapi mungkin dapat diakses melalui manipulasi parameter atau data yang dikirimkan ke server.
4. Pengujian dan Eksplorasi: Lakukan pengujian dan eksplorasi terhadap objek-objek yang berpotensi sensitif atau terbatas pada akun pengguna tertentu. Coba manipulasi parameter atau data yang dikirimkan ke server untuk memperoleh akses yang tidak sah ke objek-objek tersebut. Amati apakah ada perbedaan akses antara objek yang seharusnya tidak dapat diakses dan objek yang dapat diakses melalui IDOR.
5. Verifikasi Kekurangan Keamanan: Setelah berhasil mendapatkan akses yang tidak seharusnya melalui IDOR, verifikasi bahwa akses tersebut memang tidak sah atau tidak terotorisasi. Pastikan bahwa kerentanan IDOR ada dan bahwa akses yang diperoleh melalui manipulasi parameter atau data melanggar mekanisme kontrol akses yang seharusnya ada.
6. Dokumentasi dan Analisis: Dokumentasikan temuan secara rinci. Catat langkah-langkah yang diambil, parameter yang dimanipulasi, dan akses yang berhasil diperoleh melalui IDOR. Analisis hasil dan identifikasi dampak keamanan yang mungkin timbul dari kerentanan ini.
7. Rekomendasi dan Solusi: Berikan rekomendasi yang jelas dan konstruktif tentang bagaimana masalah IDOR dapat diperbaiki. Sertakan solusi yang memperkuat mekanisme kontrol akses, seperti menggunakan referensi terenkripsi untuk mengidentifikasi objek atau melakukan validasi akses pada sisi server.

F. Proses insecure direct object references (IDOR)

Memulai proses uji kerentanan dan celah keamanan pada Website dengan menggunakan langkah-langkah berikut:

1. Manipulasi Parameter: Teknik ini melibatkan manipulasi nilai parameter yang dikirimkan ke server untuk mengakses objek atau data yang tidak seharusnya dapat diakses. Dapat mencoba mengubah ID objek, mengganti nilai parameter dengan nilai yang tidak seharusnya diizinkan, atau menghapus parameter yang mengontrol akses.
2. Objek ID Testing: Metode ini melibatkan pengujian secara berurutan menggunakan ID yang terus bertambah atau berkurang untuk mengakses objek yang mungkin rentan terhadap IDOR. Misalnya, dapat mencoba mengakses objek dengan ID yang lebih tinggi atau lebih rendah dari ID yang seharusnya diakses.
3. Rekaman dan Pengujian Ulang: Dapat menggunakan alat rekaman untuk merekam interaksi dengan aplikasi web. Setelah merekam, sehingga dapat memodifikasi rekaman untuk mencoba mengakses objek yang tidak seharusnya diakses dan melihat respons yang diberikan oleh aplikasi.

4. Analisis Kode Sumber: Dalam beberapa kasus, perlu menganalisis kode sumber aplikasi untuk mencari tahu bagaimana pengendalian akses dan identifikasi objek dilakukan. Dengan memahami logika aplikasi, dapat mengidentifikasi potensi celah IDOR dan menguji mereka secara manual.

G. Hasil

Pada tahapan ini hasil dari deteksi Insecure Direct Object References (IDOR) diperoleh menggunakan langkah berikut:

1. Identifikasi Objek Rentan: Setelah berhasil mengidentifikasi objek atau data sensitif yang rentan terhadap serangan IDOR. Hal ini menunjukkan bahwa ada kelemahan dalam mekanisme pengendalian akses yang memungkinkan akses yang tidak sah atau tidak terotorisasi ke objek-objek tersebut.
2. Akses Tidak Sah yang Diperoleh: Jika berhasil memanipulasi parameter atau data yang dikirimkan ke server untuk mendapatkan akses yang tidak sah ke objek yang seharusnya tidak dapat diakses. Hal ini menunjukkan adanya celah keamanan IDOR yang dapat dimanfaatkan oleh penyerang.
3. Dokumentasi dan Laporan: Hasil dari deteksi IDOR perlu didokumentasikan secara rinci. Yaitu dengan mencatat langkah-langkah yang diambil, parameter yang dimanipulasi, objek yang dapat diakses, dan setiap informasi tambahan yang relevan. Hasil ini kemudian dapat disusun dalam laporan yang memberikan gambaran menyeluruh tentang kerentanan IDOR yang ditemukan.
4. Rekomendasi Perbaikan: Selain hasil deteksi IDOR, juga dapat menyertakan rekomendasi yang jelas dan konstruktif tentang bagaimana masalah dapat diperbaiki. Rekomendasi ini harus mencakup solusi teknis yang memperkuat mekanisme pengendalian akses, seperti penggunaan referensi terenkripsi, validasi akses di sisi server, atau implementasi mekanisme otentikasi yang lebih kuat.

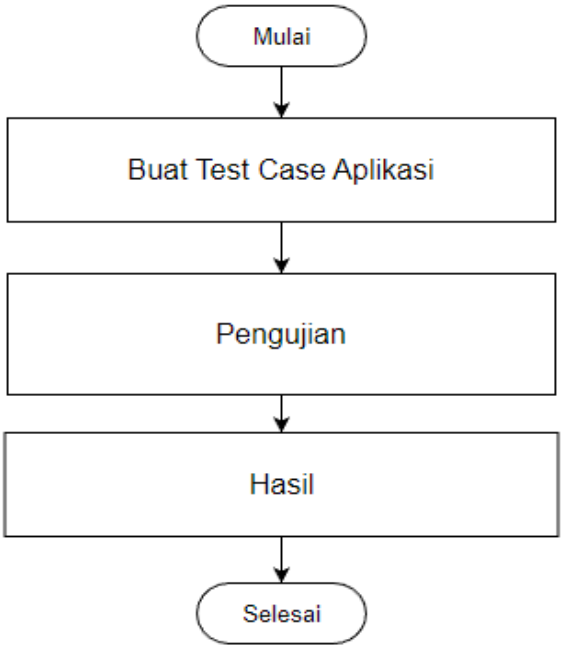
H. Pencegahan

Pada tahapan ini dilakukan untuk perbaikan bugs dan update code guna mengamankan website sebagai berikut:

1. Implementasikan Mekanisme Otentikasi dan Otorisasi yang Kuat: Pastikan aplikasi menggunakan mekanisme otentikasi yang kuat untuk mengidentifikasi pengguna dan mekanisme otorisasi yang memastikan hanya pengguna yang sah yang memiliki akses ke objek atau data sensitif.
2. Gunakan Pengidentifikasi yang Tidak Terduga: Gunakan pengidentifikasi yang tidak mudah ditebak atau ditebak oleh penyerang. Hindari menggunakan ID berturut-turut atau ID yang terkait dengan informasi pengguna lainnya yang dapat diakses secara publik.
3. Lakukan Validasi dan Kontrol Akses di Sisi Server: Pastikan validasi dan kontrol akses dilakukan di sisi server, bukan hanya di sisi klien. Ini akan membantu mencegah manipulasi parameter atau data yang dikirimkan oleh klien untuk mengakses objek yang tidak seharusnya diakses.
4. Terapkan Prinsip Least Privilege: Berikan hak akses terendah yang diperlukan untuk setiap pengguna. Hindari memberikan akses penuh atau hak akses yang tidak diperlukan pada objek atau data tertentu.
5. Enkripsi Referensi atau Identifikasi Objek: Gunakan enkripsi atau teknik hashing untuk mengamankan referensi atau identifikasi objek. Ini akan mempersulit penyerang untuk menebak atau memanipulasi referensi atau identifikasi objek yang digunakan dalam aplikasi.
6. Uji Keamanan Secara Rutin: Lakukan pengujian keamanan secara rutin untuk mengidentifikasi celah keamanan IDOR dan kerentanan lainnya, sehingga akan membantu mendeteksi dan memperbaiki masalah sebelum penyerang memanfaatkannya.

I. Perancangan sistem

Penulis menggunakan metode pengujian Black box. Metode pengujian Black box atau pengujian berdasarkan alur logika adalah metode yang fokus pada pengujian perangkat lunak dengan memahami logika atau alur kerja aplikasi tanpa memperhatikan struktur internal atau rincian implementasi dari kode sumber. Dalam metode ini, pengujian dilakukan dengan melihat masukan (input) dan keluaran (output) yang dihasilkan oleh aplikasi, tanpa mengetahui bagaimana aplikasi mencapai hasil tersebut. Pengujian dengan metode Black box dilakukan dengan menyediakan serangkaian test case yang mencakup kondisi atau perulangan tertentu dalam alur logika aplikasi. Tujuannya adalah untuk memastikan bahwa aplikasi berperilaku sesuai dengan harapan dan dapat mengatasi berbagai situasi yang mungkin terjadi.



Gambar 2 Peracangan Sistem

1. Mulai:

   Tahap ini melibatkan memulai perancangan sistem dengan mempertimbangkan keamanan sebagai faktor penting. Dalam perancangan sistem, harus mempertimbangkan arsitektur aplikasi, aliran data sensitif, dan mekanisme pengendalian akses yang diperlukan.

2. Buat Test Case Aplikasi

   Setelah merancang sistem, perlu membuat test case untuk menguji keamanan aplikasi. Dalam test case ini, memastikan untuk mencakup skenario yang mencerminkan kemungkinan serangan IDOR, termasuk mencoba mengakses objek atau data yang tidak seharusnya dapat diakses.

3. Pengujian Hasil

   Setelah test case dibuat, tahap berikutnya adalah melakukan pengujian terhadap aplikasi. Jalankan test case yang telah disusun dan periksa hasilnya. Pastikan aplikasi tidak mengalami kerentanan IDOR dan bahwa objek atau data sensitif hanya dapat diakses oleh pengguna yang berwenang.

4. Selesai

   Setelah melalui tahap pengujian dan memastikan bahwa aplikasi tahan terhadap serangan IDOR, maka dapat menyimpulkan proses perancangan sistem untuk mencegah IDOR telah selesai. Penting untuk diingat bahwa keamanan aplikasi adalah upaya berkelanjutan.

III. HASIL DAN PEMBAHASAN

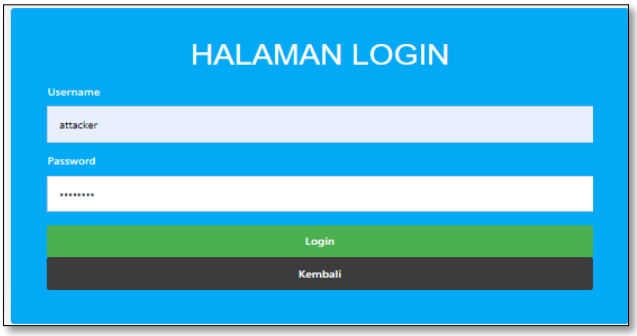
A. Persiapan data uji

Berikut adalah kegiatan uji metode IDOR yang terdiri dari 7 metode GET dan 3 metode POST sebagai berikut:

Tabel 1 Persiapan Data Uji

No.	Metode	Parameter	Objek ID	Hasil	Level Akses
1	GET	id_data	19	Tampil data dengan id_data=19 (Perencanaan Jumlah Kebutuhan BTS Dalam Penerapan M...)	Teknisi
2	GET	id_data	20	Tampil data dengan id_data=20 (VirtualBox Virtual Machine Di Linux)	Teknisi
3	GET	id_data	22	Tampil data dengan id_data=22 (Membuat Router Dengan OpenWRT)	Teknisi
4	GET	id_data	23	Tampil data dengan id_data=23 (OpenWRT: Daftar Spesifikasi Device)	Teknisi
5	GET	id_data	24	Tampil data dengan id_data=24 (Simulasi Jaringan GNS3 Menggunakan Router Mikrotik)	Teknisi
6	GET	id_data	25	Tampil data dengan id_data=25 (GNS3: Menyambungkan LAN / Switch Ke NAT)	Teknisi
7	GET	id_data	26	Tampil data dengan id_data=26 (Wajanbolic E-goen)	Teknisi
8	POST	id_users	1	Menampilkan Akun dari Teknisi Aat	Teknisi
9	POST	id_users	2	Menampilkan Akun dari Teknisi Dani Fajri	Teknisi
10	POST	id_users	3	Menampilkan Akun dari Teknisi Panjul	Teknisi

B. Percobaan insecure direct object references (IDOR) metode get

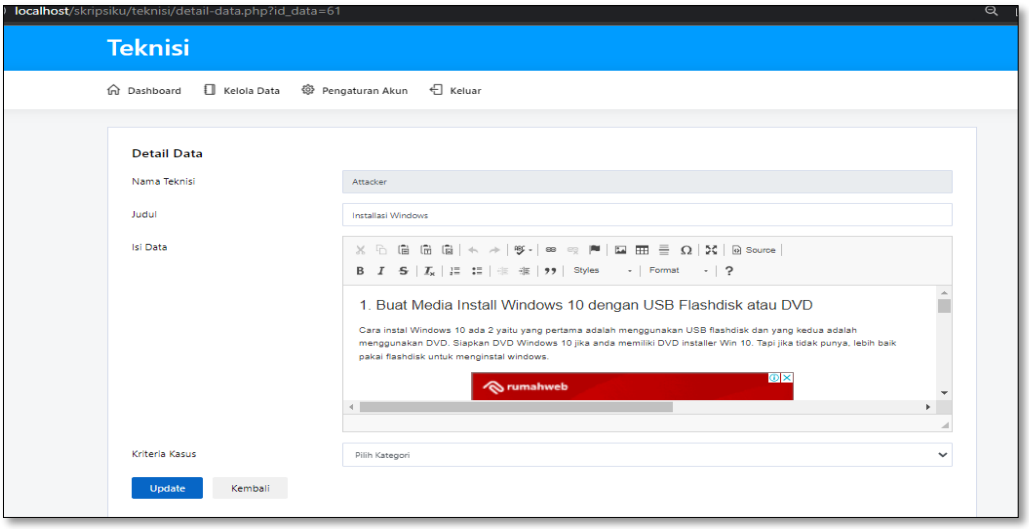


Gambar 3. Login Teknisi

Pada tahap pengujian login, tujuan utamanya adalah untuk memverifikasi apakah pengguna dengan username "attacker" dapat berhasil mengakses halaman kelola data dari teknisi lain, yang seharusnya tidak dapat diakses oleh pengguna dengan peran "attacker".

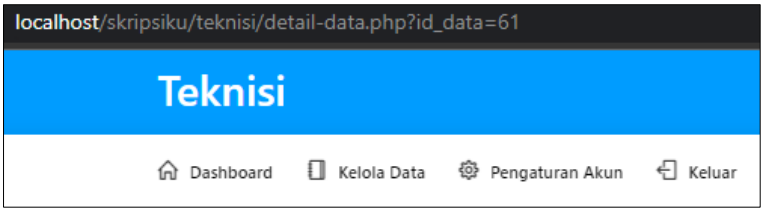
Gambar 4. Daftar Data

Dalam pengujian ini, fokusnya adalah mengedit data teknisi lain pada halaman yang menampilkan daftar data yang hanya dimiliki oleh teknisi dengan nama "Attacker". Selanjutnya, pengujian akan melibatkan upaya untuk mengedit data tersebut.



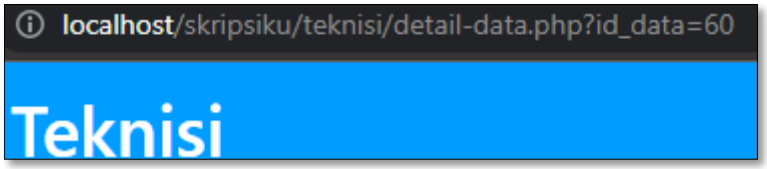
Gambar 5. Edit Data

1. Pada halaman yang menampilkan daftar data, teknisi "Attacker" melihat data dengan judul "Installasi Windows" dalam daftar tersebut.
2. Teknisi "Attacker" kemudian mencoba mengedit objek dengan mengklik tombol edit atau tindakan lainnya yang memungkinkan pengeditan data.
3. Setelah masuk ke halaman pengeditan, teknisi "Attacker" melihat bahwa ada parameter ID objek yang digunakan untuk mengidentifikasi objek yang akan diedit. URL mengandung parameter seperti "?id_data=61".
4. Teknisi "Attacker" menyadari bahwa ID objek yang terkait dengan entri "Installasi Windows" adalah 123.
5. Sebagai upaya untuk melakukan manipulasi ID objek, teknisi "Attacker" mengganti ID dalam URL menjadi ID objek yang dimilikinya atau ID objek yang seharusnya tidak dimilikinya. Teknisi "Attacker" mengganti ID menjadi 60.
6. Jika aplikasi tidak memvalidasi atau mengendalikan akses ke objek berdasarkan ID yang dimanipulasi tersebut, teknisi "Attacker" dapat mengedit objek dengan id_data=60, meskipun objek tersebut tidak seharusnya dimilikinya.
7. Dalam gambar berikutnya, mungkin terlihat bahwa teknisi "Attacker" berhasil mengedit objek dengan ID 60, yang seharusnya hanya dimiliki oleh teknisi lain.



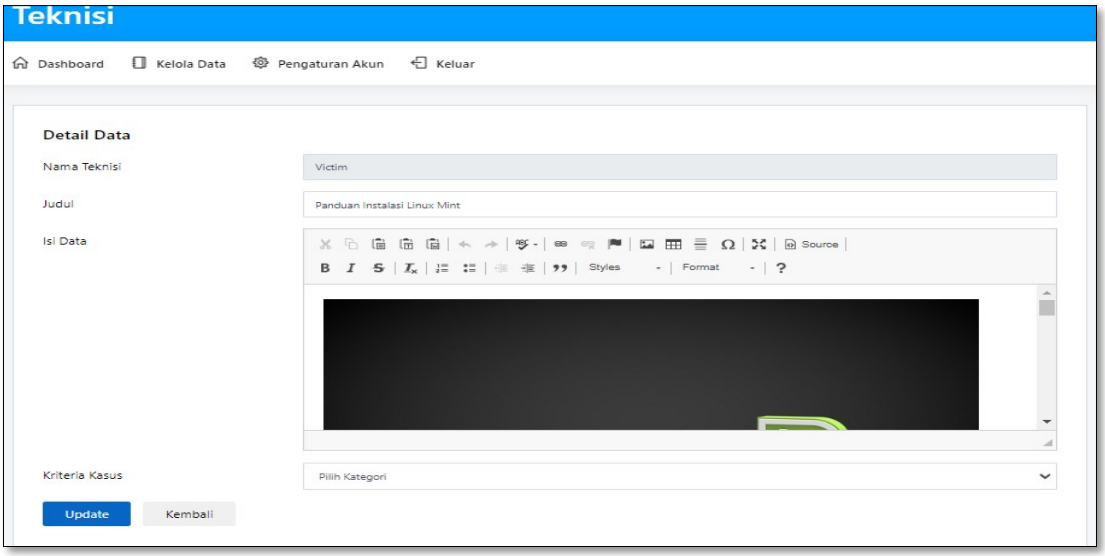
Gambar 6. Edit Objek ID

Berdasarkan gambar tersebut dirubah menjadi id_data=60 seperti berikut:



Gambar 7. Setelah Edit Objek ID

Hasilnya adalah sebagai berikut:

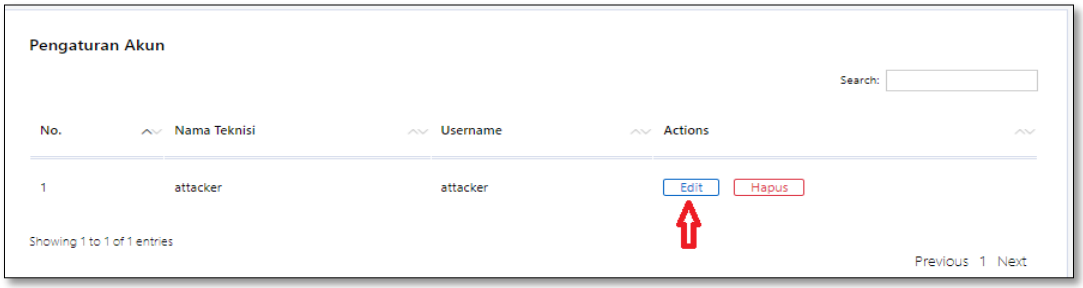


Gambar 8. Hasil Edit Objek ID

Berdasarkan gambar tersebut maka pengujian Get Data memberikan kerentanan dan membuat data dari teknisi lainnya yaitu “Victim” dapat terlihat. Dalam situasi yang aman, sistem seharusnya memberlakukan batasan akses yang tepat untuk mencegah akses tidak sah ke data sensitif. Dapat disimpulkan bahwa ada kerentanan Insecure Direct Object Reference (IDOR) yang terdeteksi pada web tersebut. IDOR terjadi ketika sebuah aplikasi tidak mengimplementasikan pengendalian akses yang memadai terhadap objek atau data sensitif yang berhubungan dengan pengguna. Hal ini memungkinkan pengguna yang tidak berhak atau pengguna dengan peran yang salah untuk mengakses atau melihat data yang tidak seharusnya mereka akses. IDOR memungkinkan teknisi dengan peran "Attacker" untuk melihat data yang seharusnya hanya dimiliki oleh teknisi dengan nama "Victim". Untuk menjaga keamanan, aplikasi web seharusnya menerapkan mekanisme pengendalian akses yang sesuai dan memastikan bahwa data hanya terlihat oleh pengguna yang berhak. Hal ini menunjukkan pentingnya mengatasi kerentanan IDOR dan melindungi data sensitif dengan mengimplementasikan pengendalian akses yang kuat dan memastikan bahwa pengguna hanya dapat mengakses data sesuai dengan hak akses yang diberikan kepada mereka.

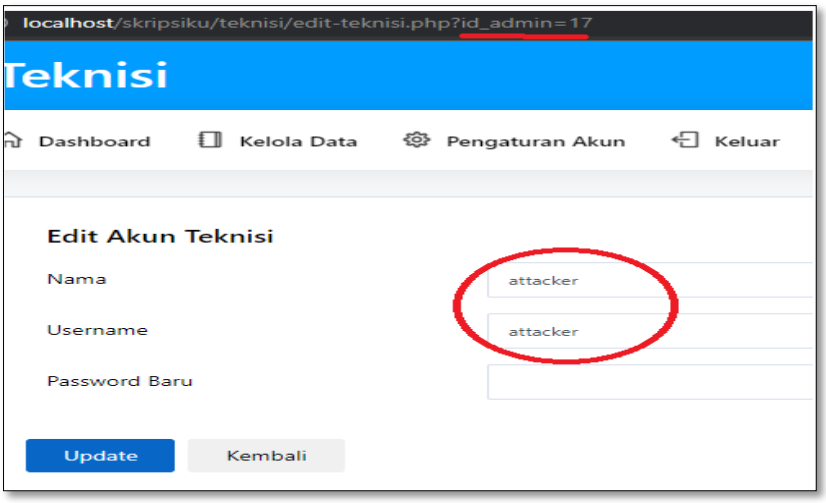
C. Percobaan insecure direct object references (IDOR) metode post

Pengujian selanjutnya adalah Metode Post, pada metode ini dilakukan pengujian dengan kasus edit password. Langkah awalnya adalah mengakses halaman pengaturan akun sebagai berikut:



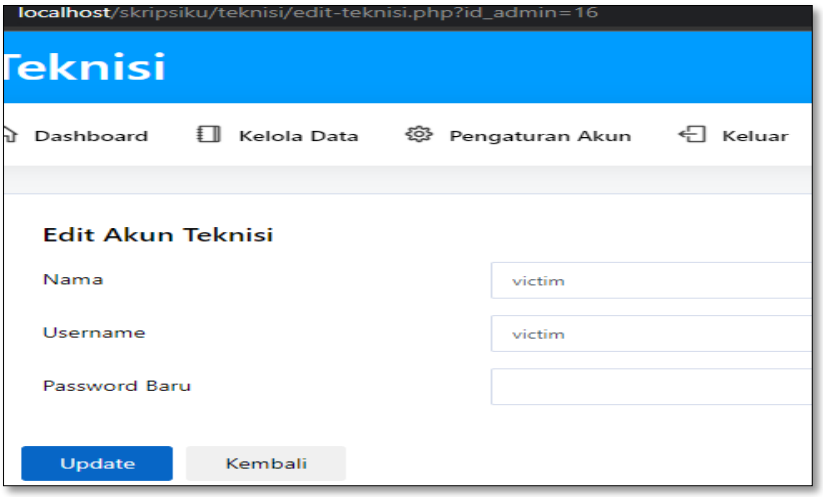
Gambar 9. Pengaturan Akun

Setelah itu klik edit akun dan menampilkan halaman berikut:



Gambar 10. Halaman Edit Akun

Berdasarkan gambar 10 dapat dilihat bahwa edit akun teknisi memiliki id_admin=17 dan jika diedit menggunakan id_admin lain hasilnya sebagai berikut:

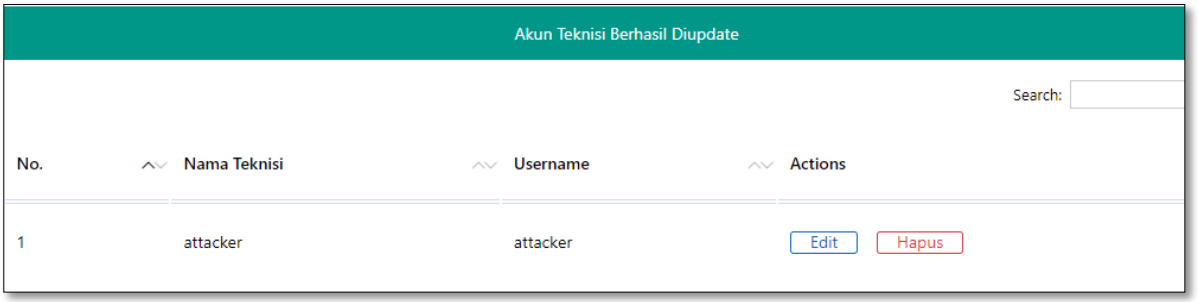


Gambar 11. Hasil Edit Objek ID

Setelah melihat bahwa dengan objek id yang diubah maka dapat memberikan informasi dari akun teknisi lain, ada kerentanan keamanan yang terdeteksi pada aplikasi yang memungkinkan akses dan pengeditan yang tidak terbatas pada akun teknisi dengan mengubah objek ID. Ini menjadi masalah keamanan yang signifikan karena dapat memberikan akses ke akun teknisi lainnya dan potensial untuk mengedit data dengan mengetahui urutan ID teknisi. Dalam kondisi yang seharusnya aman, aplikasi seharusnya menerapkan kontrol akses yang memadai untuk mencegah akses tak terbatas dan pengeditan pada akun teknisi. Namun, jika melalui manipulasi ID objek, pengguna dapat mengakses dan mengedit akun teknisi tanpa batasan, maka hal tersebut menunjukkan adanya kerentanan yang serius dalam aplikasi tersebut. Untuk menguji apakah aplikasi mampu melindungi akun teknisi dari pengeditan yang tidak sah, Pada gambar 11 dapat melihat gambar yang menunjukkan proses pengujian pengeditan akun. Dalam pengujian tersebut, apakah aplikasi memberikan batasan akses dan apakah mampu mencegah pengguna untuk melakukan pengeditan akun teknisi tanpa otorisasi yang sesuai. Selanjutnya untuk menguji edit akun dapat dilakukan atau tidak maka dapat dilihat pada gambar berikut:

Gambar 12. Uji Metode Post Edit Akun Teknisi

Berdasarkan gambar 12 yaitu dilakukan pengeditan password dan melakukan update. Sehingga hasilnya sebagai berikut:



Gambar 13. Hasil Metode Post Edit Akun

Berdasarkan Gambar 13 yang menunjukkan bahwa proses pengeditan akun teknisi lain berhasil dilakukan, dapat disimpulkan bahwa terdapat kerentanan Insecure Direct Object Reference (IDOR) yang terdeteksi pada aplikasi web tersebut. IDOR terjadi ketika aplikasi tidak menerapkan pengendalian akses yang memadai terhadap objek atau data sensitif. Dalam kasus ini, kerentanan IDOR memungkinkan pengguna untuk melakukan pengeditan pada akun teknisi lainnya, yang seharusnya tidak diizinkan. Dengan adanya kerentanan ini, dapat disimpulkan bahwa aplikasi web tersebut tidak memvalidasi dengan benar hak akses pengguna saat melakukan operasi pengeditan pada akun teknisi. Hal ini dapat mengakibatkan akses yang tidak sah terhadap data sensitif dan potensi pengubahan atau manipulasi yang tidak diinginkan.

D. Pencegahan insecure direct object references (idor) metode get dan post

Pencegahan pada detail-data.php

Code Sebelumnya:

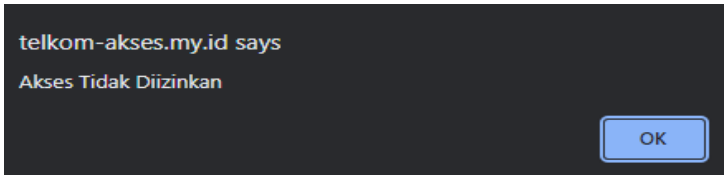
```
<?php
   $id       = mysqli_real_escape_string($connection,$_GET['id_data']);
   $query  = "SELECT * FROM tbl_data WHERE id_data ='$id'";
   $sql      = mysqli_query($connection, $query);
   $data    = mysqli_fetch_array($sql);
   //
   $query2 = "SELECT * FROM tbl_admin WHERE  id_admin='".$data['id_admin']."'";
   $sql2     = mysqli_query($connection, $query2);
   $data2   = mysqli_fetch_array($sql2);
?>
```

Menjadi:

```
<?php
   $query    = "SELECT * FROM tbl_admin WHERE username='".$_SESSION['username']."'";
   $sql       = mysqli_query($connection, $query);
   $data      = mysqli_fetch_array($sql);
   //
   $id        = mysqli_real_escape_string($connection,$_GET['id_data']);
   $query2  = "SELECT * FROM tbl_data WHERE id_data ='$id' AND id_admin ='".$data['id_admin']."'";
   $sql2      = mysqli_query($connection, $query2);
   $data2    = mysqli_fetch_array($sql2);

   if (!$data2) {
     echo "<script>alert('Akses Tidak Diizinkan');
          window.history.go(-1);
          </script>";
   }
?>
```

Pada kode tersebut ditambahkan validasi tambahan yaitu ".$_SESSION['username'] untuk mengidentifikasi akun teknisi yang login, lalu penambahan AND id_admin ='".$data['id_admin'] untuk parameter query sql serta tambahan if (!$data2) untuk memberikan informasi akses tidak diizinkan dengan seperti itu akses GET data teknisi sudah bisa dilakukan. Hasilnya sebagai berikut:



Pencegahan pada edit-teknisi.php

Code Sebelumnya:

```
<?php
   $id                = mysqli_real_escape_string($connection,$_GET['id_admin']);
   $query             = "SELECT * FROM tbl_admin WHERE id_admin ='$id'";
   $sql               = mysqli_query($connection, $query);
   $data              = mysqli_fetch_array($sql);
?>
```

Menjadi:

```
<?php
   $username          = $_SESSION['username'];
   $query             = "SELECT * FROM tbl_admin WHERE username ='$username'";
   $sql               = mysqli_query($connection, $query);
   $data              = mysqli_fetch_array($sql);
?>
```

Gambar 14. Hasil Pencegahan GET IDOR

Pada kode tersebut dilakukan perubahan teknik pemanggilan data dari objek id menjadi dari $_SESSION['username'] sehingga akun teknisi hanya ditampilkan berdasarkan akun yang digunakan saja dan tidak bisa melakukan akses ke akun teknisi lainnya.

IV. KESIMPULAN

Dalam mengakses data pada website, teknik pemanggilan data dari objek ID seringkali rentan terhadap serangan Insecure Direct Object References (IDOR). Oleh karena itu, teknik pemanggilan data dari $_SESSION['username'] dapat menjadi alternatif yang lebih aman untuk menghindari kerentanan keamanan IDOR. Dengan menggunakan teknik ini, hanya akun yang digunakan yang dapat diakses dan tidak memungkinkan akses ke akun teknisi lainnya. Penggunaan parameter query tambahan juga dapat meningkatkan keamanan website dan melindungi data serta informasi yang terdapat di dalamnya. Dengan demikian, penambahan validasi tambahan pada kode tersebut dapat membantu mencegah terjadinya kerentanan IDOR pada aplikasi web.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih yang sebesar besarnya kepada seluruh pihak-pihak yang berkontribusi dalam penelitian ini.

REFERENSI

[1] Simarmata, J., Chaerul, M., Mukti, R. C., Purba, D. W., Tamrin, A. F., Jamaludin, J., ... & Meganingratna, A. (2020). Teknologi Informasi: Aplikasi dan Penerapannya. Yayasan Kita Menulis.
[2] Primawanti, E. P., & Ali, H. (2022). Pengaruh Teknologi Informasi, Sistem Informasi Berbasis Web Dan Knowledge Management Terhadap Kinerja Karyawan (Literature Review Executive Support Sistem (Ess) for Business). Jurnal Ekonomi Manajemen Sistem Informasi, 3(3), 267-285.
[3] Guntoro, G., Costaner, L., & Musfawati, M. (2020). Analisis Keamanan Web Server Open Journal System (Ojs) Menggunakan Metode Issaf Dan Owasp (Studi Kasus Ojs Universitas Lancang Kuning). JIPI (Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika), 5(1), 45-55.
[4] Demesa, E. G. (2018). Implementation of a Hands-on Attack and Defense Lab on Insecure Direct Object References Master ' s thesis. https://www.etis.ee/Portal/Mentorships/Display/57461a3c-f3aa-40f8-a9e3-05a76e074551.
[5] Kuncoro, A. W. (2022). Pengujian Autentikasi Dan Otorisasi Web Mi-Gateway Uii Berdasarkan Dokumen Owasp Wstg V4. 2.
[6] Novendri, M. S., Saputra, A., & Firman, C. E. (2019). Aplikasi Inventaris Barang Pada Mts Nurul Islam Dumai Menggunakan Php Dan Mysql. lentera dumai, 10(2).
[7] Putri, S. E. Y. (2021). Penerapan Model Naive Bayes Untuk Memprediksi Potensi Pendaftaran Siswa Di Smk Taman Siswa Teluk Betung Berbasis Web. Journal of Engineering, Computer Science and Information Technology (JECSIT), 1(1).
[8] Kinaswara, T. A. (2019, October). Rancang Bangun Aplikasi Inventaris Berbasis Website pada Kelurahan Bantengan. In Prosiding Seminar Nasional Teknologi Informasi dan Komunikasi (SENATIK) (Vol. 2, No. 1, pp. 71-75).