JOINT CYBERSECURITY ADVISORY

Co-Authored by:

TLP:CLEAR Product ID: AA23-208A

July 27, 2023

Australian Government
Australian Signals Directorate

ACSC Australian Cyber Security Centre

Preventing Web Application Access Control Abuse

SUMMARY

The Australian Signals Directorate's Australian Cyber Security Centre (ACSC), U.S. Cybersecurity and Infrastructure Security Agency (CISA), and U.S. National Security Agency (NSA) are releasing this joint Cybersecurity Advisory to warn vendors, designers, and developers of web applications and organizations using web applications about insecure direct object reference (IDOR) vulnerabilities. IDOR vulnerabilities are access control vulnerabilities enabling malicious actors to modify or delete data or access sensitive data by issuing requests to a website or a web application programming interface (API) specifying the user identifier of other, valid users. These requests succeed where there is a failure to perform adequate authentication and authorization checks.

These vulnerabilities are frequently exploited by malicious actors in data breach incidents because they are common, hard to prevent outside the development process, and can be abused at scale. IDOR vulnerabilities have resulted in the compromise of personal, financial, and health information of millions of users and consumers.

ACSC, CISA, and NSA strongly encourage vendors, designers, developers, and end-user organizations to implement the recommendations found within the Mitigations section of this advisory—including the following—to reduce prevalence of IDOR flaws and protect sensitive data in their systems.

- **Vendors, designers, and developers** of web application frameworks and web applications: Implement secure-by-design and -default principles and ensure software performs authentication and authorization checks for every request that modifies, deletes, and accesses sensitive data.

To report suspicious or criminal activity related to information found in this Joint Cybersecurity Advisory, ***U.S organizations*** *can contact CISA's 24/7 Operations Center at report@cisa.gov or (888) 282- 0870. For NSA client requirements or general cybersecurity inquiries, contact Cybersecurity_Requests@nsa.gov. SLTT organizations should report incidents to MS-ISAC (866-787-4722 or SOC@cisecurity.org).* ***Australian organizations*** *can visit cyber.gov.au or call 1300 292 371 (1300 CYBER 1) to report cybersecurity incidents and access alerts and advisories When available, please include the following information regarding the incident: date, time, and location of the incident; type of activity; number of people affected; type of equipment used for the activity; the name of the submitting company or organization; and a designated point of contact.*

  - Use automated tools for code review to identify and remediate IDOR and other vulnerabilities.
  - Use indirect reference maps, ensuring that IDs, names, and keys are not exposed in URLs. Replace them with cryptographically strong, random values—specifically use a universally unique identifier (UUID) or a globally unique identifier (GUID).
  - Exercise due diligence when selecting third-party libraries or frameworks to incorporate into your application and keep all third-party frameworks and dependencies up to date.
- **All end-user organizations, including organizations with software-as-a-service (SaaS) models:**
  - Use due diligence when selecting web applications. Follow best practices for supply chain risk management and only source from reputable vendors.
  - Apply software patches for web applications as soon as possible.
- **End-user organizations deploying on-premises software, infrastructure-as-a-service (IaaS), or private cloud models:**
  - Review the available authentication and authorization checks in web applications that enable modification of data, deletion of data, or access to sensitive data.
  - Conduct regular, proactive vulnerability scanning and penetration testing to help ensure internet-facing web applications and network boundaries are secure.

TECHNICAL DETAILS

Description

IDOR vulnerabilities are access control vulnerabilities in web applications (and mobile phone applications [apps] using affected web API) that occur when the application or API uses an identifier (e.g., ID number, name, or key) to directly access an object (e.g., a database record) but does not properly check the authentication or authorization of the user submitting the request. Depending on the type of IDOR vulnerability, malicious actors can access sensitive data, modify or delete objects, or access functions.

- **Horizontal IDOR vulnerabilities** occur when a user can access data that they should not be able to access at the same privilege level (e.g., other user's data).
- **Vertical IDOR vulnerabilities** occur when a user can access data that they should not be able to access because the data requires a higher privilege level.
- **Object-level IDOR vulnerabilities** occur when a user can modify or delete an object that they should not be able to modify or delete.
- **Function-level IDOR vulnerabilities** occur when a user can access a function or action that they should not be able to access.

Typically, these vulnerabilities exist because an object identifier is exposed, passed externally, or easily guessed—allowing any user to use or modify the identifier.

- In **body manipulation**, an actor modifies the HTML form field data in the body of a POST request to impact targeted records.
- In **URL tampering**, an actor modifies an identifier in URLs to impact targeted records.

- In **cookie ID manipulation**, the actor modifies an identifier in a cookie to an identifier of a different user (including administrative users) in an attempt to gain access to that account.
- In **HTTP/JSON request tampering**, an actor uses a web proxy to intercept and alter arbitrary portions of legitimate requests, including values inside JSON objects.

Impact

These vulnerabilities are common[1] and hard to prevent outside the development process since each use case is unique and cannot be mitigated with a simple library or security function. Additionally, malicious actors can detect and exploit them at scale using automated tools. These factors place end-user organizations at risk of data leaks (where information is unintentionally exposed) or large-scale data breaches (where a malicious actor obtains exposed sensitive information). Data leaks or breaches facilitated by IDOR vulnerabilities include:

- An October 2021 global data leak incident where mobile phone data, including text messages, call records, photos, and geolocation from hundreds of thousands of devices was exposed by insecure “stalkerware” apps.[2] The apps collected and relayed data from the phones to the same foreign server infrastructure, which contained an IDOR vulnerability, CVE-2022-0732.[3] This led to exposure of the collected app data.[4]
- A 2019 data breach incident where more than 800 million personal financial files, including bank statements, bank account numbers, and mortgage payment documents, from a U.S. Financial Services Sector organization were exposed.[5],[6]
- A 2012 data breach incident where a malicious cyber actor obtained the personal data of more than 100,000 mobile device owners from a U.S. Communications Sector organization’s publicly accessible website.[7]

MITIGATIONS

Vendors and Developers

ACSC, CISA, and NSA recommend that vendors, designers, and implementors of web applications—including organizations that build and deploy software (such as HR tools) for their internal use and organizations that create open-source projects—implement the following mitigations. These mitigations may reduce prevalence of IDOR vulnerabilities in software and help ensure products are secure-by-design and -default.

- **Implement and inject secure-by-design and -default principles** and best practices into each stage of the software development life cycle (SDLC). Particular recommended practices are defined in the National Institute of Security and Technology’s (NIST’s) Secure Software Development Framework (SSDF), SP 800-218. Lend special attention to:
  - **Conducting code reviews** [SSDF PW 7.2, RV 1.2] against peer coding standards, checking for backdoors, malicious content, or logic flaws.
    - ACSC, CISA, and NSA recommend using automated code analysis tools for all supported releases to identify and remediate vulnerabilities**.**

- **Following secure coding practices** [SSDF PW 5.1] for web and mobile applications to ensure that they properly validate user input and generate strong user IDs.
    - **Use indirect reference maps**, such that IDs, names, and keys are not exposed in URLs. Replace them with cryptographically strong, random values—specifically use a UUID or a GUID. **Note:** UUIDs and GUIDs should not be used for security capabilities. See Request for Comment (RFC) 4122 for more information.
    - **Configure applications to deny access by default and ensure the application performs authentication and authorization checks** for every request to modify data, delete data, and access sensitive data. For example:
        - **Normalize requests**. There are many ways to encode and decode web inputs. Decode and normalize inputs before creating access control checkpoints. Ensure the access control system and other parts of the web application perform the same normalization.
        - **Implement parameter verification** leveraging syntactic and logical validation, such that web applications validate all inputs received with every HTTP/S request. Denying invalid requests can reduce the burden on the access control system.
            - Syntactic validation verifies that for each input the incoming value meets your applications' expectations. When doing syntactic validation, verify that strings are within the minimum and maximum length required, strings do not contain unacceptable characters, numeric values are within the minimum and maximum boundaries, and the input is of the proper data type.
            - Logical validation adds checks to see if the input values make sense and are consistent with design intent. When doing logical validation, verify authorization checks are performed in the correct locations, are of varying pedigree, and that there is error handling of failed authentication and authorization requests.
    - **Use CAPTCHA to limit automated invalid user requests** where feasible.
    - **Use memory-safe programming languages** where possible.
- **Testing code** to identify vulnerabilities and verify compliance with security requirements [SSDF PW 8.2].
    - Use automated testing tools to facilitate testing, fuzz testing tools to find issues with input handling,[8] and penetration testing to simulate how a threat actor may exploit the software. Consider using dynamic application security testing (DAST) tools to identify IDOR vulnerabilities in web applications.
- **Conducting role-based training** [SSDF PO 2.2] for personnel responsible for secure software development.
- **Exercising due diligence when selecting third-party libraries or frameworks** to incorporate into your application [SSDF PW 4.1].
    - Review and evaluate third-party components in the context of their expected use.
    - Verify the integrity of the product through hash or signature verification.
    - If provided, review component's Software Bill of Materials (SBOM) for outdated, vulnerable, or unauthorized applications before using it.
    - Keep all third-party frameworks and dependencies up to date to limit vulnerability inheritance. **Note:** Organizations should maintain an inventory or catalog of third-party

frameworks and dependencies to assist with proactive updates. Consider using tools to identify project dependencies and known vulnerabilities in third-party code.

For more information, see the joint Enduring Security Framework's Securing the Software Supply Chain: Recommended Practices Guide for Developers, CISA's Supply Chain Risk Management Essentials, and ACSC's Cyber Supply Chain Risk Management.

- **Establish a vulnerability disclosure program** to verify and resolve security vulnerabilities disclosed by people who may be internal or external to the organization.

Additionally, ACSC, CISA, and NSA recommend **following cybersecurity best practices** in production and enterprise environments. Software developers are high-value targets because their customers deploy software on their own trusted networks. For best practices, see:

- ACSC's Essential Eight. The Essential Eight are prioritized strategies to help cybersecurity professionals mitigate cybersecurity incidents caused by various cyber threats.
- CISA's Cross-Sector Cybersecurity Performance Goals (CPGs). The CPGs, developed by CISA and NIST, are a prioritized subset of IT and OT security practices that can meaningfully reduce the likelihood and impact of known cyber risks and common tactics, techniques, and procedures. Because the CPGs are a subset of best practices, ACSC, CISA, and NSA also recommend software manufacturers implement a comprehensive information security program based on a recognized framework, such as the NIST Cybersecurity Framework (CSF).
- NSA's Top Ten Cybersecurity Mitigations. The Top Ten sets priorities for enterprise activities to counter a broad range of exploitation techniques and minimize mission impact.

All End-User Organizations

ACSC, CISA, and NSA recommend that all end-user organizations, including those with on-premises software, SaaS, IaaS, and private cloud models, implement the mitigations below to improve their cybersecurity posture.

- **Exercise due diligence when selecting web applications**. Follow best practices for supply chain risk management and source from reputable vendors that demonstrate commitment to secure-by-design and -default principles.
    - Verify the integrity of the product through hash or signature verification.
    - If provided, review the SBOM for outdated, vulnerable, or unauthorized applications before using the product.

  For more information, see the Enduring Security Framework's Securing the Software Supply Chain: Recommended Practices Guide for Customers, CISA's Supply Chain Risk Management Essentials, and ACSC's Cyber Supply Chain Risk Management.

- **Apply software patches for web applications** as soon as possible.
- **Configure the application to log and generate alerts from tamper attempts**—with this information, network defenders can investigate and take appropriate follow-on actions.

    - Establish a baseline to efficiently identify abnormal behavior. **Note:** Web application error codes such as `HTTP 404` and `HTTP 403` are associated with common enumeration techniques.
    - Aggregate logs into a centralized solution (e.g., a security information and event management [SIEM] tool) to facilitate active monitoring and threat hunting.
- **Create, maintain, and exercise a basic cyber incident response plan (IRP) and associated communications plan**. Plans should include response and notification procedures for data breach and cyber incidents. For more information, see:
    - ACSC: Preparing for and Responding to Cyber Incidents
    - ACSC: Cyber Incident Response Plan - Guidance
    - ACSC: Cyber Incident Response Readiness Checklist
    - Office of the Australian Information Commissioner (OAIC): Data Breach Preparation and Response
    - OIAC: Data Breach Response Plan
    - CISA: Incident Response Plan Basics
    - CISA: Federal Government Cybersecurity Incident and Vulnerability Response Playbook (Although tailored to U.S. Federal Civilian Branch (FCEB) agencies, these playbooks provide operational procedures for planning and conducting cybersecurity incident and vulnerability response activities and detail steps for both incident and vulnerability response.)
    - CISA: Protecting Sensitive and Personal Information from Ransomware-Caused Data Breaches

Additionally, ACSC, CISA, and NSA recommend **following cybersecurity practices**. For best practices, see ACSC's Essential Eight, CISA's CPGs, and NSA's Top Ten Cybersecurity Mitigation Strategies.

End-User Organizations with On-Premises Software, IaaS, or Private Cloud Models

ACSC, CISA, and NSA recommend that organizations:

- **Conduct regular, proactive penetration testing** to ensure network boundaries, as well as web applications, are secure. Prioritize web applications that are internet-facing and contain user login functionality. Such testing may be beyond the technical or financial capabilities of some organizations. Consider using a trusted third party for penetration testing to discover new attack vectors (notably prior to deployment of new or altered internet-facing services). **Note:** Organizations should consult with their legal counsel as appropriate to determine which systems and applications can be included in the scope of the penetration testing.
    - **Use web application penetration testing tools** to capture the user identifier sent to the web server when requesting a web page containing sensitive data and map all locations where user input is used to reference objects directly. Test with users of various privilege levels (e.g., a normal user and admin user).
- **Use DAST and other vulnerability scanners** to detect IDOR vulnerabilities. DAST tools identify vulnerabilities in web applications with penetration tests and generate automated

alerts. **Note:** Exercise due diligence when selecting DAST tools. Not all DAST tools can detect IDOR vulnerabilities—tools with the ability may need the environment configured in a specific way and may also need custom rules in place. Sufficient DAST tools often ingest the application API documentation to build a model of the application. While these tools can be used to detect IDOR vulnerabilities, they are not foolproof and should be used with other security testing methods to ensure comprehensive coverage.

- **Immediately report detected vulnerabilities to the vendor or developer**. Alternatively (or if the vendor or developer fails to respond), report the vulnerability to CISA at cisa.gov/report.
- **Consider establishing a vulnerability disclosure program** to verify, resolve, and report security vulnerabilities disclosed by people who may be internal or external to the organization.
- **Use a web application firewall (WAF)** to filter, monitor, and block malicious HTTP/S traffic traveling to the web application.
- **Use a data loss prevention (DLP) tool to** prevent unauthorized data from leaving the application.

ACSC, CISA, and NSA recommend that organizations with on-premises software or IaaS consider using SaaS models for their internet-facing websites.

End-User Organizations with SaaS Models

Organizations leveraging SaaS with sufficient resources may consider conducting penetration testing and using vulnerability scanners. However, such tests may interfere with service provider operations. Organizations should consult with their legal counsel as appropriate to determine what can be included in the scope of the penetration testing.

INCIDENT RESPONSE

If you or your organization are victim to a data breach or cyber incident, follow relevant cyber incident response and communications plans, as appropriate.

- **Australia:** Australian organizations that have been impacted or require assistance in regards to a cybersecurity incident can contact ACSC via 1300 CYBER1 (1300 292 371), or by submitting a report to cyber.gov.au.
- **United States:** U.S. organizations may report cybersecurity incidents to CISA's 24/7 Operations Center at Report@cisa.dhs.gov, cisa.gov/report, or (888) 282-0870. When available, please include the information regarding the incident: date, time, and location of the incident; type of activity; number of people affected; type of equipment used for the activity; the name of the submitting company or organization; and a designated point of contact.

RESOURCES

- For additional guidance on designing secure-by-design and -default products, See joint guide Shifting the Balance of Cybersecurity Risk: Principles and Approaches for Security-by-Design and -Default.
- For additional guidance on protecting against data breaches, see ACSC's webpage on data breaches.

REFERENCES

[1] A01 Broken Access Control - OWASP Top 10:2021

[2] A massive 'stalkerware' leak puts the phone data of thousands at risk

[3] Mobile device monitoring services do not authenticate API requests

[4] Behind the stalkerware network spilling the private phone data of hundreds of thousands

[5] First American Financial Corp. Leaked Hundreds of Millions of Title Insurance Records

[6] Biggest Data Breaches in US History [Updated 2023]

[7] AT&T Hacker 'Weev' Sentenced to 3.5 Years in Prison

[8] Fuzzing | OWASP Foundation

DISCLAIMER

The information in this report is being provided "as is" for informational purposes only. ACSC, CISA, and NSA do not endorse any commercial product or service, including any subjects of analysis. Any reference to specific commercial products, processes, or services by service mark, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement, recommendation, or favoring by the United States or Australian Governments, and this guidance shall not be used for advertising or product endorsement purposes.

PURPOSE

This document was developed in furtherance of the authors' cybersecurity missions, including their responsibilities to identify and disseminate threats, and to develop and issue cybersecurity specifications and mitigations. This information may be shared broadly to reach all appropriate stakeholders.

VERSION HISTORY

July 27, 2023: Initial version.