**A blue planet in space

Description automatically generated**A logo with a triangle and a blue triangle

Description automatically generatedA cartoon of an astronaut holding a computer

Description automatically generated

A black background with white text

Description automatically generatedA black background with white text

Description automatically generated

**PENETRATION TESTING REPORT**

|  |
| --- |
| **Acme Corp Portal** |

ZDEMO9999

Sunday, July 7, 2024

**DOCUMENT CONTROL**

|  |  |
| --- | --- |
| **AUTHOR(S)** | **Hugo Chun-Li** |
| **REVIEWER** | **Ryu Ken** |
| **APPROVER** | **Thomas A. Anderson** |

**VERSION HISTORY**

|  |  |  |  |
| --- | --- | --- | --- |
| **VERSION** | **DESCRIPTION** | **DATE** | **STATUS** |
| **0.1** | **Initial Draft** | **7/5/24** | **DRAFT** |
| **0.2** | **Revisions** | **7/6/24** | **DRAFT** |
| **1.0** | **Release** | **7/7/24** | **RELEASE** |

**PROJECT TEAM**

|  |  |  |  |
| --- | --- | --- | --- |
| **TEAM MEMBER** | **JOB TITLE** | **EMAIL** | **PROJECT ROLE** |
| **Hugo Chun-Li** | **Senior Penetration Tester** | **hugo.chun-li@attackforge.com** | **Pentester** |
| **Thomas A. Anderson** | **Security Testing Manager** | **Thomas.A.Anderson@attackforge.com** | **Project Manager** |
| **Ryu Ken** | **Principal Information Security Consultant** | **ryu.ken@attackforge.com** | **Pentest Lead** |
| **Blanka M. Bison** | **Senior Software Engineer** | **blanka.bison@attackforge.com** | **Engineer** |

TABLE OF CONTENTS

[**EXECUTIVE SUMMARY** 5](#_Toc171286818)

[TESTING PROGRESS 5](#_Toc171286819)

[UNIQUE VULNERABILITIES 5](#_Toc171286820)

[REMEDIATION PROGRESS 5](#_Toc171286821)

[OVERVIEW 6](#_Toc171286822)

[**TESTING SUMMARY** 7](#_Toc171286823)

[BACKGROUND 7](#_Toc171286824)

[APPROACH 7](#_Toc171286825)

[METHODOLOGY 7](#_Toc171286826)

[SCOPE 7](#_Toc171286827)

[OUT OF SCOPE 8](#_Toc171286828)

[CUSTOMER GOALS 8](#_Toc171286829)

[TESTING TEAM GOALS 8](#_Toc171286830)

[ASSUMPTIONS AND CONSTRAINTS 8](#_Toc171286831)

[**TESTING OUTCOME** 9](#_Toc171286832)

[SUMMARY OF RECOMMENDATIONS 9](#_Toc171286833)

[POSITIVE SECURITY OBSERVATIONS 9](#_Toc171286834)

[**RETESTING HISTORY** 10](#_Toc171286835)

[**SUMMARY FINDINGS** 11](#_Toc171286836)

[**ATTACKCHAINS** 13](#_Toc171286837)

[1. Gain control of core web server to further pivot attack into Client internal network. 13](#_Toc171286838)

[**VULNERABILITIES** 14](#_Toc171286839)

[1. Unrestricted Upload of File with Dangerous Type 14](#_Toc171286840)

[2. Inconsistent Access Control 18](#_Toc171286841)

[3. Relative Path Traversal 20](#_Toc171286842)

[4. Reflected Cross Site Scripting 22](#_Toc171286843)

[5. Blind SQL Injection 25](#_Toc171286844)

[6. Persistent Cross Site Scripting 27](#_Toc171286845)

[7. Unsupported Web Server Detection 29](#_Toc171286846)

[8. PHP Unsupported Version Detection 31](#_Toc171286847)

[9. SSL Version 2 and 3 Protocol Detection 33](#_Toc171286848)

[10. Weak Password Policy 36](#_Toc171286849)

[11. Session Fixation 38](#_Toc171286850)

[12. SSL Weak Ciphers 40](#_Toc171286851)

[13. Reflected Cross Site Scripting 41](#_Toc171286852)

[14. Cross Site Request Forgery (CSRF) 44](#_Toc171286853)

[15. Functionality Misuse 46](#_Toc171286854)

[16. No Account Lockout May Facilitate Brute Force Password Attack 48](#_Toc171286855)

[17. Open Redirection 50](#_Toc171286856)

[18. SSL Certificate Expiry 52](#_Toc171286857)

[19. SSL Weak Cipher Suites Supported 54](#_Toc171286858)

[20. SSL Certificate Signed Using Weak Hashing Algorithm 56](#_Toc171286859)

[21. SSL Medium Strength Cipher Suites Supported (SWEET32) 58](#_Toc171286860)

[22. SSL Certificate Cannot Be Trusted 61](#_Toc171286861)

[23. SSL Self-Signed Certificate 64](#_Toc171286862)

[24. Multiple Simultaneous User Sessions 66](#_Toc171286863)

[25. HTML 5 Cross Origin Resource Sharing 68](#_Toc171286864)

[26. Missing X-XSS-Protection Header 70](#_Toc171286865)

[27. Cookie Without HTTPOnly Flag Set 71](#_Toc171286866)

[28. Strict Transport Security Policy Not Enforced 73](#_Toc171286867)

[29. Outdated Version of JQuery 75](#_Toc171286868)

[30. Cookie With Secure Flag Missing 76](#_Toc171286869)

[31. User Enumeration 78](#_Toc171286870)

[32. Server Discloses Supporting Technology 80](#_Toc171286871)

[33. Insufficient Framing Protection 81](#_Toc171286872)

[34. Insufficient Session Expiration 83](#_Toc171286873)

[35. SSL RC4 Cipher Suites Supported (Bar Mitzvah) 84](#_Toc171286874)

[36. Inducing Account Lockout 86](#_Toc171286875)

[37. TLS Version 1.1 Protocol Detection 88](#_Toc171286876)

[38. nginx HTTP Server Detection 90](#_Toc171286877)

[39. TLS Version 1.0 Protocol Detection 91](#_Toc171286878)

[**TEST CASES** 93](#_Toc171286879)

[**UNIQUE VULNERABILITY DETAILS** 102](#_Toc171286880)

# **EXECUTIVE SUMMARY**

|  |  |  |  |
| --- | --- | --- | --- |
| TESTING PROGRESS | | | |
| **Start** | **Jan 02 2024** | **Tested** | **98.59%** |
| **End** | **Jan 06 2024** | **In Progress** | **0%** |
| **Completed** | **100%** | **Not Tested** | **0%** |
| **Total Testcases** | **142** | **Not Applicable** | **1.41%** |
| UNIQUE VULNERABILITIES | | | |
| **Total** | **39** | image | |
| **Critical** | **1** |
| **High** | **8** |
| **Medium** | **14** |
| **Low** | **12** |
| **Info** | **4** |
| REMEDIATION PROGRESS | | | |
| **Open** | **40** | image | |
| **Retest** | **4** |
| **Closed** | **5** |
| OVERVIEW | | | |
| Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam ipsum augue, finibus id pretium sit amet, semper nec lorem. Nam bibendum arcu sed odio iaculis, a pretium eros ornare. Suspendisse in imperdiet ipsum, sed congue nisi. In ullamcorper feugiat bibendum. Vestibulum eu diam sed diam ultrices cursus eu vel nisl. Phasellus vestibulum est eu faucibus tempor. Integer id elementum enim, quis tristique est.   1. **Integer neque urna, elementum at nibh ut, ultrices pulvinar lacus.** 2. **Mauris ut convallis magna. Curabitur id felis odio.** 3. **Fusce felis risus, imperdiet vitae orci nec, consequat rutrum tortor.**   image Figure 1: Compromising the Domain Controller..  Ut mi nulla, finibus et est at, malesuada mollis purus. Nam sollicitudin urna in pharetra sagittis. Duis fringilla tincidunt nulla, eu accumsan nulla vestibulum at. Proin non consectetur purus. Nam ut commodo sem. Vestibulum sodales dolor eget elit volutpat ornare. | | | |

# 

# **TESTING SUMMARY**

|  |
| --- |
| BACKGROUND |
| The objective of testing was to assess the security posture for the in-scope internal network ranges for ACME Corp., from the perspective of an unauthenticated user connected to the network(s).  Testing was performed between <START> to <END> from <LOCATION>.  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam ipsum augue, finibus id pretium sit amet, semper nec lorem. Nam bibendum arcu sed odio iaculis, a pretium eros ornare. Suspendisse in imperdiet ipsum, sed congue nisi. In ullamcorper feugiat bibendum. Vestibulum eu diam sed diam ultrices cursus eu vel nisl. Phasellus vestibulum est eu faucibus tempor. Integer id elementum enim, quis tristique est. |
| APPROACH |
| The following scenario's were assessed:   * Attacker has breached physical security of Building A, Floor 9 and plugged into the network. * Attacker has compromised general staff VPN service. |
| METHODOLOGY |
| The workflow of this assessment included the following phases:  **1. Reconnaissance**  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam ipsum augue, finibus id pretium sit amet, semper nec lorem. Nam bibendum arcu sed odio iaculis, a pretium eros ornare. Suspendisse in imperdiet ipsum, sed congue nisi. In ullamcorper feugiat bibendum. Vestibulum eu diam sed diam ultrices cursus eu vel nisl. Phasellus vestibulum est eu faucibus tempor. Integer id elementum enim, quis tristique est.  **2. Enumeration**  Integer neque urna, elementum at nibh ut, ultrices pulvinar lacus. Donec eget turpis porttitor lectus laoreet euismod. Vivamus suscipit gravida metus vitae pellentesque. Ut maximus dictum mi, ut accumsan nulla maximus et. Vestibulum auctor quis nulla pulvinar eleifend. Aliquam aliquam iaculis est blandit dapibus. Sed posuere ipsum sed consectetur ultrices. Vestibulum feugiat vulputate magna eget commodo.  **3. Exploitation**  Mauris ut convallis magna. Curabitur id felis odio. Cras molestie porttitor ante, ut hendrerit sapien rutrum ut. Quisque eget fringilla turpis. Ut luctus est orci, auctor viverra nunc mattis non. Phasellus nec lacus at enim scelerisque varius id pharetra quam. Donec eget odio magna. Praesent ante metus, tincidunt eu tellus sit amet, suscipit venenatis mi. Sed ut venenatis metus. Sed faucibus dignissim pulvinar. Ut et urna risus. Curabitur vel faucibus ipsum.  **4. Documentation**  Fusce felis risus, imperdiet vitae orci nec, consequat rutrum tortor. Ut magna felis, tempus in purus sit amet, consequat porttitor ante. In dapibus tempor scelerisque. Integer et tellus in ante sodales varius. Ut mi nulla, finibus et est at, malesuada mollis purus. Nam sollicitudin urna in pharetra sagittis. Duis fringilla tincidunt nulla, eu accumsan nulla vestibulum at. Proin non consectetur purus. Nam ut commodo sem. Vestibulum sodales dolor eget elit volutpat ornare. |
| SCOPE |
| classic.acmegroup.co, form.acmegroup.co, globexcorp.com.au |
| OUT OF SCOPE |
| The following items considered out-of-scope for this assessment:   * Any APIs following /integrations/... * Testing from the Admin user role * ... |
| CUSTOMER GOALS |
| * Make sure to test for X on the user profile section * Make sure to include Y when testing ABC * Get good coverage on 123 |
| TESTING TEAM GOALS |
| * Identify Critical and High risk issues particularly for X. * Review ABC to determine if any issues. * Determine whether the design of 123 is compliant with HIPAA |
| ASSUMPTIONS AND CONSTRAINTS |
| The following assumptions have been taken into consideration:   1. Lorem ipsum dolor sit amet, consectetur adipiscing elit. 2. Integer neque urna, elementum at nibh ut, ultrices pulvinar lacus. 3. Mauris ut convallis magna. Curabitur id felis odio. |

# **TESTING OUTCOME**

|  |
| --- |
| SUMMARY OF RECOMMENDATIONS |
| The following recommendations are made to <CUSTOMER> as a result of this assessment:  **Recommendation 1: Do this thing..**  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam ipsum augue, finibus id pretium sit amet, semper nec lorem. Nam bibendum arcu sed odio iaculis, a pretium eros ornare. Suspendisse in imperdiet ipsum, sed congue nisi. In ullamcorper feugiat bibendum. Vestibulum eu diam sed diam ultrices cursus eu vel nisl. Phasellus vestibulum est eu faucibus tempor. Integer id elementum enim, quis tristique est.  **Recommendation 2: Do that thing..**  Integer neque urna, elementum at nibh ut, ultrices pulvinar lacus. Donec eget turpis porttitor lectus laoreet euismod. Vivamus suscipit gravida metus vitae pellentesque. Ut maximus dictum mi, ut accumsan nulla maximus et. Vestibulum auctor quis nulla pulvinar eleifend. Aliquam aliquam iaculis est blandit dapibus. Sed posuere ipsum sed consectetur ultrices. Vestibulum feugiat vulputate magna eget commodo. |
| POSITIVE SECURITY OBSERVATIONS |
| The following positive security observations were observed during this assessment:   * Lorem ipsum dolor sit amet, consectetur adipiscing elit. * Nullam ipsum augue, finibus id pretium sit amet, semper nec lorem. * Nam bibendum arcu sed odio iaculis, a pretium eros ornare. * Suspendisse in imperdiet ipsum, sed congue nisi. |

# **RETESTING HISTORY**

|  |  |  |  |
| --- | --- | --- | --- |
| **ROUND** | **TEST WINDOW** | **STATUS** | **RETESTED** |
| 4 | 5/2/19 - 5/4/19 | Completed | Relative Path Traversal |
| 3 | 4/28/19 - 4/29/19 | Completed | Cookie With Secure Flag Missing |
| 2 | 4/25/19 - 4/27/19 | Completed | Inconsistent Access Control  Persistent Cross Site Scripting |
| 1 | 4/5/19 - 4/9/19 | Completed | Session Fixation  Strict Transport Security Policy Not Enforced |

# 

# **SUMMARY FINDINGS**

|  |  |  |
| --- | --- | --- |
| **PRIORITY** | **VULNERABILITY** | **STATUS** |
| **CRITICAL** | **Unrestricted Upload of File with Dangerous Type** | **OPEN** |
| **HIGH** | **Inconsistent Access Control** | **OPEN** |
| **HIGH** | **Relative Path Traversal** | **OPEN** |
| **HIGH** | **Blind SQL Injection** | **OPEN** |
| **HIGH** | **SSL Version 2 and 3 Protocol Detection** | **OPEN** |
| **HIGH** | **Reflected Cross Site Scripting** | **CLOSED** |
| **HIGH** | **Persistent Cross Site Scripting** | **CLOSED** |
| **HIGH** | **Unsupported Web Server Detection** | **CLOSED** |
| **HIGH** | **PHP Unsupported Version Detection** | **CLOSED** |
| **MEDIUM** | **Weak Password Policy** | **OPEN** |
| **MEDIUM** | **Session Fixation** | **OPEN** |
| **MEDIUM** | **SSL Weak Ciphers** | **OPEN** |
| **MEDIUM** | **Reflected Cross Site Scripting** | **OPEN** |
| **MEDIUM** | **Cross Site Request Forgery (CSRF)** | **OPEN** |
| **MEDIUM** | **Functionality Misuse** | **OPEN** |
| **MEDIUM** | **No Account Lockout May Facilitate Brute Force Password Attack** | **OPEN** |
| **MEDIUM** | **Open Redirection** | **OPEN** |
| **MEDIUM** | **SSL Certificate Expiry** | **OPEN** |
| **MEDIUM** | **SSL Weak Cipher Suites Supported** | **OPEN** |
| **MEDIUM** | **SSL Certificate Signed Using Weak Hashing Algorithm** | **OPEN** |
| **MEDIUM** | **SSL Medium Strength Cipher Suites Supported (SWEET32)** | **OPEN** |
| **MEDIUM** | **SSL Certificate Cannot Be Trusted** | **OPEN** |
| **MEDIUM** | **SSL Self-Signed Certificate** | **OPEN** |
| **LOW** | **Multiple Simultaneous User Sessions** | **OPEN** |
| **LOW** | **HTML 5 Cross Origin Resource Sharing** | **OPEN** |
| **LOW** | **Missing X-XSS-Protection Header** | **OPEN** |
| **LOW** | **Cookie Without HTTPOnly Flag Set** | **OPEN** |
| **LOW** | **Strict Transport Security Policy Not Enforced** | **OPEN** |
| **LOW** | **Outdated Version of JQuery** | **OPEN** |
| **LOW** | **Cookie With Secure Flag Missing** | **OPEN** |
| **LOW** | **User Enumeration** | **OPEN** |
| **LOW** | **Server Discloses Supporting Technology** | **OPEN** |
| **LOW** | **Insufficient Framing Protection** | **OPEN** |
| **LOW** | **Insufficient Session Expiration** | **OPEN** |
| **LOW** | **SSL RC4 Cipher Suites Supported (Bar Mitzvah)** | **OPEN** |
| **INFO** | **Inducing Account Lockout** | **OPEN** |
| **INFO** | **TLS Version 1.1 Protocol Detection** | **OPEN** |
| **INFO** | **nginx HTTP Server Detection** | **OPEN** |
| **INFO** | **TLS Version 1.0 Protocol Detection** | **OPEN** |

# **ATTACKCHAINS**

**Attack Objective**

## Gain control of core web server to further pivot attack into Client internal network.

|  |
| --- |
| A person wearing a black hoodie and using a computer  Description automatically generated |
| **External Attacker**  Attacker who has self-registered account on Client Internet-facing application. |
| **A black arrow on a black background  Description automatically generated with medium confidence** |
| A person in a black hoodie using a computer  Description automatically generated |
| **Action**  Log into application and enumerate vulnerable file-upload functionality within the application. |
| **A black arrow on a black background  Description automatically generated with medium confidence** |
| A person wearing a black hoodie and gloves  Description automatically generated |
| **Exploit Critical Vulnerability**  Attacker identifies vulnerable 'upload Avatar' functionality in MyProfile and uploads web shell. |
| **A black arrow on a black background  Description automatically generated with medium confidence** |
| A person wearing a black hoodie and gloves  Description automatically generated |
| **Exploit High Vulnerability**  Attacker enumerates server directory structure to navigate directly to uploaded web shell. |
| **A black arrow on a black background  Description automatically generated with medium confidence** |
|  |
| **Target Server**  Attacker triggers web shell, elevates to full shell, then creates back door in web server for persistent remote access. |
| **A black arrow on a black background  Description automatically generated with medium confidence** |
|  |
| **Captured Flag**  Operating-System access to compromised Client web server allowing further attack into Client internal network. |
|  |

# **VULNERABILITIES**

|  |  |  |
| --- | --- | --- |
|  | Unrestricted Upload of File with Dangerous Type |  |
| **CVSSv3 SCORE** | | |
| Base: **9.8**  Temporal: **9.8**  Environmental: **9.8**  Vector: **CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H** | | |
| **DESCRIPTION** | | |
| **An unrestricted upload of files with dangerous type** often occurs in applications that explicitly trust application users will submit files of specific type and content only. Uploaded files are then either passed to other internal components for further processing or remain stored for future use within the application in an easily guessable location.  **In a typical attack scenario**, an attacker discovers the upload functionality in the application and submits a specifically crafted file that embeds malicious content, e.g. virus, exploit or shell code. From this point the attacker either passively waits until the malicious content is accessed and executed by an internal component, other system or user, or if the file is stored in a discoverable location the attacker tries triggering file execution within application by leveraging application mapping of known file types to specific execution routines. An insider or an attacker who can get administrative access to application can upload a web shell, then upload a normal shell and escalate privileges. The attack can be further propagated to other hosts on same network segment. | | |
| **ATTACK SCENARIO** | | |
| Arbitrary code execution is possible if an uploaded file is interpreted and executed as code by the recipient. This is especially true for .asp and .php extensions uploaded to web servers because these file types are often treated as automatically executable, even when file system permissions do not specify execution.  image Figure 2 | | |
| **REMEDIATION RECOMMENDATION** | | |
| 1. Assume all input is malicious. Use an 'accept known good' input validation strategy, i.e. use a whitelist of acceptable inputs that strictly conform to specifications. Reject any input that does not strictly conform to specifications, or transform it into something that does. 2. Generate a new, unique filename for an uploaded file instead of using the user-supplied filename, so that no external input is used at all. 3. Define a very limited set of allowable extensions and only generate filenames that end in these extensions. 4. Consider the possibility of XSS (CWE-79) before allowing .html or .htm file types. 5. Ensure that only one extension is used in the filename. Some web servers, including some versions of Apache, may process files based on inner extensions so that 'filename.php.gif' is fed to the PHP interpreter. 6. For any security checks that are performed on the client side, ensure that these checks are duplicated on the server side, in order to avoid CWE-602. Attackers can bypass the client-side checks by modifying values after the checks have been performed, or by changing the client to remove the client-side checks entirely. Then, these modified values would be submitted to the server. 7. Do not rely exclusively on the MIME content type or filename attribute when determining how to render a file. Validating the MIME content type and ensuring that it matches the extension is only a partial solution. 8. Do not rely exclusively on looking for malicious or malformed inputs (i.e., do not rely on a blacklist). A blacklist is likely to miss at least one undesirable input, especially if the code's environment changes. This can give attackers enough room to bypass the intended validation. However, blacklists can be useful for detecting potential attacks or determining which inputs are so malformed that they should be rejected outright. 9. For example, limiting filenames to alphanumeric characters can help to restrict the introduction of unintended file extensions. | | |
| **AFFECTED ASSETS** | | |
| **Open** - **globexcorp.com.au**  **REMEDIATION NOTES**  *09/27/2023*  Engineering is working on addressing this issue in the following sections of the app:   * /api/users/profile * /api/users/avatar * /api/company/logo   image Figure 3: Fixed in common component  **NOTES**  During testing, it was possible to gain system-level privileges for the application server (through uploading of web shell) which could then be used to create persistent connection to pivot attack towards other internal systems/hosts.  **PROOF OF CONCEPT / STEPS TO REPRODUCE**  How to reproduce   1. Do this... 2. Do that...   <some script>...<do something>...</some script>  image Figure 4 | | |
| **EVIDENCE** | | |
| None. | | |

|  |  |  |
| --- | --- | --- |
|  | Inconsistent Access Control |  |
| **CVSSv3 SCORE** | | |
| Base: **8.9**  Temporal: **8.9**  Environmental: **8.9**  Vector: **CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:C/C:H/I:H/A:L** | | |
| **DESCRIPTION** | | |
| It appears that the application does not consistently apply access controls to all its resources. The lack of access protection for some sensitive resources can be leveraged by an non-authorised attacker to either gather important information for a consequent attack against other application users, or to access and modify directly unprotected application data.  Assuming a user with a given identity, authorisation is the process of determining whether that user can access a given resource, based on the user's privileges and any permissions or other access-control specifications that apply to the resource. When access control checks are not applied consistently - or not at all - users are able to access data or perform actions that they should not be allowed to perform. This can lead to a wide range of problems, including information exposures, denial of service, and arbitrary code execution. | | |
| **ATTACK SCENARIO** | | |
| The page can be identified quick and easily through application fingerprinting and crawling. An attacker could read sensitive data, either by reading the data directly from a data store that is not properly restricted, or by accessing insufficiently-protected, privileged functionality to read the data. | | |
| **REMEDIATION RECOMMENDATION** | | |
| This issue should be fixed by applying proper authorisation permission to the affected resources unless it is not an intended business feature.  For web applications, make sure that the access control mechanism is enforced correctly at the server side on every page. Users should not be able to access any unauthorised functionality or information by simply requesting direct access to that page. One way to do this is to ensure that all pages containing sensitive information are not cached, and that all such pages restrict access to requests that are accompanied by an active and authenticated session token associated with a user who has the required permissions to access that page.  Ensure that you perform access control checks related to your business logic. These checks may be different than the access control checks that you apply to more generic resources such as files, connections, processes, memory, and database records. For example, a database may restrict access for medical records to a specific database user, but each record might only be intended to be accessible to the patient and the patient's doctor.  Reduce the attack surface by carefully mapping roles with data and functionality. Use role-based access control (RBAC) to enforce the roles at the appropriate boundaries. Note that this approach may not protect against horizontal authorisation, i.e., it will not protect a user from attacking others with the same role. | | |
| **AFFECTED ASSETS** | | |
| **Open** - **globexcorp.com.au**  **REMEDIATION NOTES**  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  **PROOF OF CONCEPT / STEPS TO REPRODUCE**  - Login in to the application as Standard user  - Once in the dashboard, take note and copy the value of the 'X' visible from within the URL  - Note that the administrative tab does not appear when clicking on the icon on the top right-hand side as an standard user role is used  - Replace the value of 'X' in the following URL with the copied one and browse to it:globexcorp.com.au/crm/test/page?X=6365973057477497141868253726 | | |
| **EVIDENCE** | | |
| image  screenshot.png | | |

|  |  |  |
| --- | --- | --- |
|  | Relative Path Traversal |  |
| **CVSSv3 SCORE** | | |
| Base: **7.5**  Temporal: **7.5**  Environmental: **7.5**  Vector: **CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:C/C:N/I:L/A:H** | | |
| **DESCRIPTION** | | |
| The software uses external input to construct a pathname that should be within a restricted directory, but it does not properly neutralize sequences such as '..' that can resolve to a location that is outside of that directory.  This allows attackers to traverse the file system to access files or directories that are outside of the restricted directory. | | |
| **ATTACK SCENARIO** | | |
| The attacker may be able to create or overwrite critical files that are used to execute code, such as programs or libraries. The attacker may be able to overwrite or create critical files, such as programs, libraries, or important data. If the targeted file is used for a security mechanism, then the attacker may be able to bypass that mechanism. For example, appending a new account at the end of a password file may allow an attacker to bypass authentication. The attacker may be able read the contents of unexpected files and expose sensitive data. If the targeted file is used for a security mechanism, then the attacker may be able to bypass that mechanism.   For example, by reading a password file, the attacker could conduct brute force password guessing attacks in order to break into an account on the system. The attacker may be able to overwrite, delete, or corrupt unexpected critical files such as programs, libraries, or important data. This may prevent the software from working at all and in the case of a protection mechanisms such as authentication, it has the potential to lockout every user of the software. | | |
| **REMEDIATION RECOMMENDATION** | | |
| Assume all input is malicious. Use an 'accept known good' input validation strategy, i.e., use a whitelist of acceptable inputs that strictly conform to specifications. Reject any input that does not strictly conform to specifications, or transform it into something that does.  When performing input validation, consider all potentially relevant properties, including length, type of input, the full range of acceptable values, missing or extra inputs, syntax, consistency across related fields, and conformance to business rules. As an example of business rule logic, 'boat' may be syntactically valid because it only contains alphanumeric characters, but it is not valid if the input is only expected to contain colours such as 'red' or 'blue.'   Do not rely exclusively on looking for malicious or malformed inputs (i.e., do not rely on a blacklist). A blacklist is likely to miss at least one undesirable input, especially if the code's environment changes. This can give attackers enough room to bypass the intended validation. However, blacklists can be useful for detecting potential attacks or determining which inputs are so malformed that they should be rejected outright. When validating filenames, use stringent whitelists that limit the character set to be used. If feasible, only allow a single '.' character in the filename to avoid weaknesses such as CWE-23, and exclude directory separators such as '/' to avoid CWE-36. Use a whitelist of allowable file extensions, which will help to avoid CWE-434. Do not rely exclusively on a filtering mechanism that removes potentially dangerous characters. This is equivalent to a blacklist, which may be incomplete (CWE-184).   For example, filtering '/' is insufficient protection if the filesystem also supports the use of '' as a directory separator. Another possible error could occur when the filtering is applied in a way that still produces dangerous data (CWE-182). For example, if '../' sequences are removed from the '.../...//' string in a sequential fashion, two instances of '../' would be removed from the original string, but the remaining characters would still form the '../' string. | | |
| **AFFECTED ASSETS** | | |
| **Open** - **globexcorp.com.au**  **REMEDIATION NOTES**  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  **PROOF OF CONCEPT / STEPS TO REPRODUCE**  Preconditions  Authenticate to the portal as administrator user and browse to the following URL:  https://globexcorp.com.au/test/cmsedit.jsp?file=../../../../../../../../etc/hostname  Note that the resulting page will contain the hostname of the underlying system.  Affected   * URL: https://globexcorp.com.au/test/cmsedit.jsp?file=../../../../../../../../etc/hostname * Parameter: file | | |
| **EVIDENCE** | | |
| None. | | |

|  |  |  |
| --- | --- | --- |
|  | Reflected Cross Site Scripting |  |
| **CVSSv3 SCORE** | | |
| Base: **7.9**  Temporal: **7.9**  Environmental: **7.9**  Vector: **CVSS:3.1/AV:N/AC:H/PR:L/UI:R/S:C/C:L/I:H/A:H** | | |
| **DESCRIPTION** | | |
| Cross Site Scripting  Cross-site scripting (XSS) vulnerability occurs when data submitted to the application is not properly handled before being embedded within the applications response or stored for later retrieval.  Reflected cross-site scripting  Reflected cross-site scripting (XSS) occurs when a server receives data directly from a HTTP request and returns (or reflects) it back in the HTTP response. In a typical XSS attack scenario, exploitation takes place when an attacker causes a victim to supply dangerous content to a vulnerable web application, which is then reflected back to the victim and executed by the web browser.  **The most common mechanism for delivering malicious content** is to include it as a parameter in a URL that is posted publicly or e-mailed directly to the victim. URLs constructed in this manner constitute the core of many phishing schemes, whereby an attacker convinces a victim to visit a URL that refers to a vulnerable site. After the site reflects the attacker's content back to the victim, the content is executed by the victim's browser.  **The most common attack** performed with XSS involves the disclosure of session or other sensitive information stored in user cookies. Typically, a malicious user will craft a client-side script, which when parsed by a web browser performs some activity (such as sending all site cookies to a given e-mail address). This script will be loaded and run by each user visiting the vulnerable component of the web site. Since the site requesting to run the script has access to the cookies in question, the malicious script does also. For example, an attacker could redirect users to malicious web sites.  More sophisticated attacks may extend to, for example, an attacker using advanced XSS exploitation tools like the Browser Exploitation Framework (BeEF). | | |
| **ATTACK SCENARIO** | | |
| XSS injection attack is a well-documented attack with a number of automated tools available to facilitate discovery, exploitation and post-exploitation control processes. XSS can cause a variety of problems for the end user that range in severity from an annoyance to complete account compromise. Some XSS vulnerabilities can be exploited to manipulate or steal cookies, create requests that can be mistaken for those of a valid user, compromise confidential information, or execute malicious code on the end user systems for a variety of nefarious purposes. Other damaging attacks include the disclosure of end user files, installation of Trojan horse programs, redirecting the user to some other page or site, running 'Active X' controls (under Microsoft Internet Explorer) from sites that a user perceives as trustworthy, and modifying presentation of content. An attack against the larger user base of the application may result in successful compromise of users computers and potential infection with malware that would effectively allow further compromise of users data.  image Figure 5 | | |
| **REMEDIATION RECOMMENDATION** | | |
| To prevent XSS attacks a multi-layered approach is recommended.   * Input received from the client should be strictly validated on the server side before any further processing takes place. * The filter should use a White List approach by only accepting Known Good characters. * Validation should be performed on a per field basis and should endeavour to be as strict as possible. * Ensure that data is fully normalised and decoded before being compared to the filter. * All client supplied data should be HMTL encoded at the point where it is displayed to the user. This includes request data such as query string parameters and data retrieved from storage. * It is recommended that all alphanumeric characters be HTML encoded to avoid XSS. However the following characters must be encoded: double quotes, ampersand, less than sign, and greater than sign | | |
| **AFFECTED ASSETS** | | |
| **Closed** - **globexcorp.com.au**  **REMEDIATION NOTES**  *06/20/2024*  Issue Closed: Issue has been fixed  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  **PROOF OF CONCEPT / STEPS TO REPRODUCE**  1. do this..  2. do this.. | | |
| **EVIDENCE** | | |
| None. | | |

|  |  |  |
| --- | --- | --- |
|  | Blind SQL Injection |  |
| **CVSSv3 SCORE** | | |
| Base: **8**  Temporal: **8**  Environmental: **8.1**  Vector: **CVSS:3.1/AV:N/AC:H/PR:L/UI:R/S:C/C:H/I:H/A:H** | | |
| **DESCRIPTION** | | |
| The application suffers from systemic pre and post authentication blind SQL injection issues. This issue is a result of the application not conducting thorough sanitisation of user-supplied data prior to being included in SQL queries. This also indicates that queries are generated through string concatenation rather than using parameterisation (parameterised queries). An attacker may include (or inject) any content into the final query, through manipulation of the vulnerable page parameters. This results in an attacker being able to issue arbitrary commands to the database. | | |
| **ATTACK SCENARIO** | | |
| Through exploitation of this issue it could be possible to obtain information from the database, regardless of the web application's authorisation controls. This includes sensitive information such as the usernames and passwords of other application users. It is also possible to manipulate or delete data present within the database. The total amount of data lost would be determined by the effectiveness of organisation's backup policy. | | |
| **REMEDIATION RECOMMENDATION** | | |
| Modify the application to validate all user input before using it in a database query. Furthermore, modify the application to use parameterised database queries, instead of constructing queries through string concatenation. User input should undergo a validation process, the strongest of which is regarded as whitelisting, which ensures that provided data adheres to a set of rules. This is commonly performed via the use of regular expressions or other pattern matching technologies, which can specify string rules regarding permitted content types. Escaping is often performed in order to sanitise quote symbols which are used within the SQL query to denote particular values. This process is conducted transparently when parameterised queries are used. Where string concatenation is used to develop a query, this process must be manually conducted. It is important to note that quote escaping is not sufficient to prevent all SQL injection vectors. In particular, user-supplied data is often used for values which aren't encapsulated within quotes. Examples of which include ORDER BY and LIMIT. | | |
| **AFFECTED ASSETS** | | |
| **Open** - **globexcorp.com.au**  **REMEDIATION NOTES**  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  **PROOF OF CONCEPT / STEPS TO REPRODUCE**  It is possible to verify the existence of this issue by visiting the following URL and observing the that the page returned the application for user 597098 successfully:  - https://globexcorp.com.au/crm/test/ReportBase.jsp?UserId=597098%20and%203=3  Note the result won't include the report anymore. This is because the logic operand injected in the SQL query is AND 3=1 which returns false. In turn the database will return false to the application and did not return any data. This is a simple example that demonstrates the logic of SQL injection and operands. | | |
| **EVIDENCE** | | |
| None. | | |

|  |  |  |
| --- | --- | --- |
|  | Persistent Cross Site Scripting |  |
| **CVSSv3 SCORE** | | |
| Base: **7.6**  Temporal: **7.6**  Environmental: **7.6**  Vector: **CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:L/I:H/A:L** | | |
| **DESCRIPTION** | | |
| Cross-site scripting (XSS) vulnerability occurs when data submitted to the application is not properly handled before being embedded within the applications response or stored for later retrieval. Persistent XSS attacks are those where the injected script is permanently stored on the target servers, such as in a database, in a message forum files, visitor log, comment field, etc. A victim retrieves the malicious script from the server when the victim requests the stored information.   The most common attack performed with XSS involves the disclosure of session or other sensitive information stored in user cookies. Typically, a malicious user will craft a client-side script, which when parsed by a web browser performs some activity (such as sending all site cookies to a given e-mail address). This script will be loaded and run by each user visiting the vulnerable component of the web site. Since the site requesting to run the script has access to the cookies in question, the malicious script does also. For example, an attacker could load this as an invisible iframe in any sites under his control to silently infect users who would view the affected profile in their browsers.   More sophisticated attacks may extend to, for example, an attacker using advanced XSS exploitation tools like The Browser Exploitation Framework (BeEF) in order to target corporate users or administrators of the application. | | |
| **ATTACK SCENARIO** | | |
| XSS injection attack is a well-documented attack. A number of automated tools are publically available to facilitate discovery, exploitation and post-exploitation phases of the attack. The identified instance of persistent XSS exists in the public interface, no privileged access is required in order to exploit the vulnerability. If successful the attack can lead to a range of scenarios that depend on type of victim a candidate, a corporate user (e.g. HR user) or an administrator. In a worst case scenario the attack can result in a compromise of client network. | | |
| **REMEDIATION RECOMMENDATION** | | |
| To prevent XSS attacks a multi-layered defence approach is recommended. Input received from the client should be strictly validated on the server side before any further processing takes place. The filter should use a White List approach by only accepting Known Good characters. Validation should be performed on a per field basis and should endeavour to be as strict as possible. Ensure that data is fully normalised and decoded before being compared to the filter. All client supplied data should be HMTL encoded at the point where it is displayed to the user. This includes request data such as query string parameters and data retrieved from storage. It is recommended that all alphanumeric characters be HTML encoded to avoid XSS. However the following characters must be encoded: double quotes, ampersand, less than sign, and greater than sign. | | |
| **AFFECTED ASSETS** | | |
| **Closed** - **globexcorp.com.au**  **REMEDIATION NOTES**  *06/20/2024*  Issue Closed: Issue has been fixed  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  **PROOF OF CONCEPT / STEPS TO REPRODUCE**  To reproduce this issue, open an existing or create a new application from the 'New Application' menu. Then access the Notes section from the submenu on the right hand side of the page. Fill the subject field with 'test' and the message field with 'alert('XSS')' then save. The page will automatically reload resulting in a a pop up message with 'XSS' as text. This type of JavaScript code is benign, however an attacker would use malicious code that can attack other users of the application, for example to steal their session cookie/token. | | |
| **EVIDENCE** | | |
| None. | | |

|  |  |  |
| --- | --- | --- |
|  | Unsupported Web Server Detection |  |
| **CVSSv3 SCORE** | | |
| Base: **10**  Temporal: **10**  Environmental: **10**  Vector: **CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H** | | |
| **DESCRIPTION** | | |
| According to its version, the remote web server is obsolete and no longer maintained by its vendor or provider.  Lack of support implies that no new security patches for the product will be released by the vendor. As a result, it may contain security vulnerabilities.  The remote web server is obsolete / unsupported. | | |
| **ATTACK SCENARIO** | | |
| N/A | | |
| **REMEDIATION RECOMMENDATION** | | |
| Remove the service if it is no longer needed. Otherwise, upgrade to a newer version if possible or switch to another server. | | |
| **AFFECTED ASSETS** | | |
| **Closed** - **form.acmegroup.co**  **REMEDIATION NOTES**  *06/20/2024*  Issue Closed: Issue has been fixed  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  **PROOF OF CONCEPT / STEPS TO REPRODUCE**  Product : Apache 2.2.x  Server response header : Apache/2.2.34 (Amazon)  Supported versions : Apache HTTP Server 2.4.x  Additional information : http://archive.apache.org/dist/httpd/Announcement2.2.html  **Closed** - **classic.acmegroup.co**  **REMEDIATION NOTES**  *06/20/2024*  Issue Closed: Issue has been fixed  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  **PROOF OF CONCEPT / STEPS TO REPRODUCE**  Product : Apache 2.2.x  Server response header : Apache/2.2.34 (Amazon)  Supported versions : Apache HTTP Server 2.4.x  Additional information : http://archive.apache.org/dist/httpd/Announcement2.2.html | | |
| **EVIDENCE** | | |
| None. | | |

|  |  |  |
| --- | --- | --- |
|  | PHP Unsupported Version Detection |  |
| **CVSSv3 SCORE** | | |
| Base: **8.1**  Temporal: **8.1**  Environmental: **8.1**  Vector: **CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:H** | | |
| **DESCRIPTION** | | |
| According to its version, the installation of PHP on the remote host is no longer supported.  Lack of support implies that no new security patches for the product will be released by the vendor. As a result, it is likely to contain security vulnerabilities.  The remote host contains an unsupported version of a web application scripting language. | | |
| **ATTACK SCENARIO** | | |
| N/A | | |
| **REMEDIATION RECOMMENDATION** | | |
| Upgrade to a version of PHP that is currently supported. | | |
| **AFFECTED ASSETS** | | |
| **Closed** - **classic.acmegroup.co**  **REMEDIATION NOTES**  *06/20/2024*  Issue Closed: Issue has been fixed  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  **PROOF OF CONCEPT / STEPS TO REPRODUCE**  Source : X-Powered-By: PHP/5.3.29  Installed version : 5.3.29  End of support date : 2014/08/14  Announcement : http://php.net/archive/2014.php#id2014-08-14-1  Supported versions : 7.1.x / 7.2.x / 7.3.x | | |
| **EVIDENCE** | | |
| None. | | |

|  |  |  |
| --- | --- | --- |
|  | SSL Version 2 and 3 Protocol Detection |  |
| **CVSSv3 SCORE** | | |
| Base: **7.5**  Temporal: **7.5**  Environmental: **7.5**  Vector: **CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N** | | |
| **DESCRIPTION** | | |
| The remote service accepts connections encrypted using SSL 2.0 and/or SSL 3.0. These versions of SSL are affected by several cryptographic flaws, including:   - An insecure padding scheme with CBC ciphers.   - Insecure session renegotiation and resumption schemes.  An attacker can exploit these flaws to conduct man-in-the-middle attacks or to decrypt communications between the affected service and clients.  Although SSL/TLS has a secure means for choosing the highest supported version of the protocol (so that these versions will be used only if the client or server support nothing better), many web browsers implement this in an unsafe way that allows an attacker to downgrade a connection (such as in POODLE). Therefore, it is recommended that these protocols be disabled entirely.  NIST has determined that SSL 3.0 is no longer acceptable for secure communications. As of the date of enforcement found in PCI DSS v3.1, any version of SSL will not meet the PCI SSC's definition of 'strong cryptography'.  The remote service encrypts traffic using a protocol with known weaknesses. | | |
| **ATTACK SCENARIO** | | |
| N/A | | |
| **REMEDIATION RECOMMENDATION** | | |
| Consult the application's documentation to disable SSL 2.0 and 3.0. Use TLS 1.1 (with approved cipher suites) or higher instead. | | |
| **AFFECTED ASSETS** | | |
| **Open** - **classic.acmegroup.co**  **REMEDIATION NOTES**  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  **PROOF OF CONCEPT / STEPS TO REPRODUCE**  - SSLv3 is enabled and the server supports at least one cipher.  Explanation: TLS 1.0 and SSL 3.0 cipher suites may be used with SSLv3    Medium Strength Ciphers (> 64-bit and < 112-bit key, or 3DES)   EDH-RSA-DES-CBC3-SHA Kx=DH Au=RSA Enc=3DES-CBC(168) Mac=SHA1   ECDHE-RSA-DES-CBC3-SHA Kx=ECDH Au=RSA Enc=3DES-CBC(168) Mac=SHA1   DES-CBC3-SHA Kx=RSA Au=RSA Enc=3DES-CBC(168) Mac=SHA1    High Strength Ciphers (>= 112-bit key)   DHE-RSA-AES128-SHA Kx=DH Au=RSA Enc=AES-CBC(128) Mac=SHA1   DHE-RSA-AES256-SHA Kx=DH Au=RSA Enc=AES-CBC(256) Mac=SHA1   DHE-RSA-CAMELLIA128-SHA Kx=DH Au=RSA Enc=Camellia-CBC(128) Mac=SHA1   DHE-RSA-CAMELLIA256-SHA Kx=DH Au=RSA Enc=Camellia-CBC(256) Mac=SHA1   DHE-RSA-SEED-SHA Kx=DH Au=RSA Enc=SEED-CBC(128) Mac=SHA1   ECDHE-RSA-AES128-SHA Kx=ECDH Au=RSA Enc=AES-CBC(128) Mac=SHA1   ECDHE-RSA-AES256-SHA Kx=ECDH Au=RSA Enc=AES-CBC(256) Mac=SHA1   AES128-SHA Kx=RSA Au=RSA Enc=AES-CBC(128) Mac=SHA1   AES256-SHA Kx=RSA Au=RSA Enc=AES-CBC(256) Mac=SHA1   CAMELLIA128-SHA Kx=RSA Au=RSA Enc=Camellia-CBC(128) Mac=SHA1   CAMELLIA256-SHA Kx=RSA Au=RSA Enc=Camellia-CBC(256) Mac=SHA1   SEED-SHA Kx=RSA Au=RSA Enc=SEED-CBC(128) Mac=SHA1   DHE-RSA-AES128-SHA256 Kx=DH Au=RSA Enc=AES-CBC(128) Mac=SHA256   DHE-RSA-AES256-SHA256 Kx=DH Au=RSA Enc=AES-CBC(256) Mac=SHA256   ECDHE-RSA-AES128-SHA256 Kx=ECDH Au=RSA Enc=AES-CBC(128) Mac=SHA256   ECDHE-RSA-AES256-SHA384 Kx=ECDH Au=RSA Enc=AES-CBC(256) Mac=SHA384   RSA-AES128-SHA256 Kx=RSA Au=RSA Enc=AES-CBC(128) Mac=SHA256   RSA-AES256-SHA256 Kx=RSA Au=RSA Enc=AES-CBC(256) Mac=SHA256   The fields above are :   {OpenSSL ciphername}  Kx={key exchange}  Au={authentication}  Enc={symmetric encryption method}  Mac={message authentication code}  {export flag} | | |
| **EVIDENCE** | | |
| None. | | |

|  |  |  |
| --- | --- | --- |
|  | Weak Password Policy |  |
| **CVSSv3 SCORE** | | |
| Base: **6.8**  Temporal: **6.8**  Environmental: **6.8**  Vector: **CVSS:3.1/AV:N/AC:L/PR:L/UI:R/S:U/C:L/I:H/A:L** | | |
| **DESCRIPTION** | | |
| The application was found to implement a weak password policy. The application does not require that users should have strong passwords, which makes it easier for attackers to compromise user accounts. An authentication mechanism is only as strong as its credentials. For this reason, it is important to require users to have strong passwords. Lack of password complexity significantly reduces the search space when trying to guess user's passwords, making brute-force attacks easier. | | |
| **ATTACK SCENARIO** | | |
| The complexity rules specified by the application will allow users to create passwords that are easily recovered during brute force and offline password attacks. An attacker could easily guess user passwords and gain access to user accounts. | | |
| **REMEDIATION RECOMMENDATION** | | |
| Configure the application to require passwords that conform to a strong complexity policy by increasing entropy. This can be achieved by enforcing minimum character length and enforcing the use of uppercase characters, numbers and special characters in passwords.  A password strength policy should contain the following attributes:  - Minimum and maximum length;  - Require mixed character sets (alpha, numeric, special, mixed case);  - Do not contain user name;  - Expiration; and - No password reuse.  Authentication mechanisms should always require sufficiently complex passwords and require that they be periodically changed. | | |
| **AFFECTED ASSETS** | | |
| **Open** - **globexcorp.com.au**  **REMEDIATION NOTES**  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  *01/06/2024*  Issue Re-Opened: Issue has not been fixed  **PROOF OF CONCEPT / STEPS TO REPRODUCE**  Through the admin panel, change a user's password to a known weak password such as 'password' or '1234'. Take note that the password change is successful. The login form enforces the use of passwords that are longer than 8 characters, however this restriction only applies to the input box on the login form and not the authentication endpoint. Therefore it is possible to submit a valid password of less than 8 characters to the authentication endpoint and be successfully logged in, making brute force attacks possible against weak passwords. | | |
| **EVIDENCE** | | |
| None. | | |

|  |  |  |
| --- | --- | --- |
|  | Session Fixation |  |
| **CVSSv3 SCORE** | | |
| Base: **5**  Temporal: **5**  Environmental: **5**  Vector: **CVSS:3.1/AV:N/AC:H/PR:L/UI:N/S:U/C:L/I:L/A:L** | | |
| **DESCRIPTION** | | |
| Authenticating a user, or otherwise establishing a new user session, without invalidating any existing session identifier, gives an attacker the opportunity to steal authenticated sessions. Such a scenario is commonly observed when:  - A web application authenticates a user without first invalidating the existing session, thereby continuing to use the session already associated with the user.  - An attacker is able to force a known session identifier on a user so that, once the user authenticates, the attacker has access to the authenticated session. - The application or container uses predictable session identifiers. In the generic exploit of session fixation vulnerabilities, an attacker creates a new session on a web application and records the associated session identifier. The attacker then causes the victim to associate, and possibly authenticate, against the server using that session identifier, giving the attacker access to the user's account through the active session. | | |
| **ATTACK SCENARIO** | | |
| An attacker who gain's access to an existing session token will be able to gain privileges and assume identity of the affected user. | | |
| **REMEDIATION RECOMMENDATION** | | |
| Invalidate any existing session identifiers prior to authorising a new user session. Ensure sessions are properly terminated when a user logs out, or after a defined period of inactivity.  For platforms such as ASP that do not generate new values for sessionid cookies, utilise a secondary cookie. In this approach, set a secondary cookie on the user's browser to a random value and set a session variable to the same value. If the session variable and the cookie value ever don't match, invalidate the session, and force the user to log on again. | | |
| **AFFECTED ASSETS** | | |
| **Open** - **globexcorp.com.au**  **REMEDIATION NOTES**  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  **PROOF OF CONCEPT / STEPS TO REPRODUCE**  1. Using an intercepting proxy or browser development tools, login as user 'X' and generate a case report.  2. Take note of the value for the 'X' cookie.  3. Log out and clear all browser cache.  4. Repeat step 1.  5. Take note the value for the 'X' cookie is same as previous i.e. static. | | |
| **EVIDENCE** | | |
| None. | | |

|  |  |  |
| --- | --- | --- |
|  | SSL Weak Ciphers |  |
| **CVSSv3 SCORE** | | |
| Base: **5**  Temporal: **5**  Environmental: **5**  Vector: **CVSS:3.1/AV:N/AC:H/PR:N/UI:R/S:U/C:L/I:L/A:L** | | |
| **DESCRIPTION** | | |
| The remote host supports the use of SSL ciphers that offer medium strength encryption. The security community regards medium strength as any encryption that uses key lengths at least 64 bits and less than 128 bits, uses the RC4/3DES encryption suite, or has SSL 3.0 or TLSv1 enabled. | | |
| **ATTACK SCENARIO** | | |
| An attacker could exploit this issue to potentially decrypt and view application data transmitted by the affected system to an application user or application component. Note that it is considerably easier to circumvent medium strength encryption if the attacker is on the same physical network. | | |
| **REMEDIATION RECOMMENDATION** | | |
| Reconfigure the affected application if possible to avoid use of medium strength ciphers. Disable the cryptographically insecure RC4/3DS ciphers from the web server configuration and ensure that the OpenSSL library is up to date. Disable support for SSLv3 and TLSv1. | | |
| **AFFECTED ASSETS** | | |
| **Open** - **globexcorp.com.au**  **REMEDIATION NOTES**  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  *01/06/2024*  Issue Re-Opened: Issue has not been fixed  **PROOF OF CONCEPT / STEPS TO REPRODUCE**  Use a tool such ass sslscan to enumerate the ciphers supported by the application server. | | |
| **EVIDENCE** | | |
| None. | | |

|  |  |  |
| --- | --- | --- |
|  | Reflected Cross Site Scripting |  |
| **CVSSv3 SCORE** | | |
| Base: **4**  Temporal: **4**  Environmental: **4**  Vector: **CVSS:3.1/AV:N/AC:H/PR:H/UI:R/S:C/C:L/I:N/A:L** | | |
| **DESCRIPTION** | | |
| Cross Site Scripting  Cross-site scripting (XSS) vulnerability occurs when data submitted to the application is not properly handled before being embedded within the applications response or stored for later retrieval.  Reflected cross-site scripting  Reflected cross-site scripting (XSS) occurs when a server receives data directly from a HTTP request and returns (or reflects) it back in the HTTP response. In a typical XSS attack scenario, exploitation takes place when an attacker causes a victim to supply dangerous content to a vulnerable web application, which is then reflected back to the victim and executed by the web browser.  **The most common mechanism for delivering malicious content** is to include it as a parameter in a URL that is posted publicly or e-mailed directly to the victim. URLs constructed in this manner constitute the core of many phishing schemes, whereby an attacker convinces a victim to visit a URL that refers to a vulnerable site. After the site reflects the attacker's content back to the victim, the content is executed by the victim's browser.  **The most common attack** performed with XSS involves the disclosure of session or other sensitive information stored in user cookies. Typically, a malicious user will craft a client-side script, which when parsed by a web browser performs some activity (such as sending all site cookies to a given e-mail address). This script will be loaded and run by each user visiting the vulnerable component of the web site. Since the site requesting to run the script has access to the cookies in question, the malicious script does also. For example, an attacker could redirect users to malicious web sites.  More sophisticated attacks may extend to, for example, an attacker using advanced XSS exploitation tools like the Browser Exploitation Framework (BeEF). | | |
| **ATTACK SCENARIO** | | |
| XSS injection attack is a well-documented attack with a number of automated tools available to facilitate discovery, exploitation and post-exploitation control processes. XSS can cause a variety of problems for the end user that range in severity from an annoyance to complete account compromise. Some XSS vulnerabilities can be exploited to manipulate or steal cookies, create requests that can be mistaken for those of a valid user, compromise confidential information, or execute malicious code on the end user systems for a variety of nefarious purposes. Other damaging attacks include the disclosure of end user files, installation of Trojan horse programs, redirecting the user to some other page or site, running 'Active X' controls (under Microsoft Internet Explorer) from sites that a user perceives as trustworthy, and modifying presentation of content. An attack against the larger user base of the application may result in successful compromise of users computers and potential infection with malware that would effectively allow further compromise of users data.  image Figure 6 | | |
| **REMEDIATION RECOMMENDATION** | | |
| To prevent XSS attacks a multi-layered approach is recommended.   * Input received from the client should be strictly validated on the server side before any further processing takes place. * The filter should use a White List approach by only accepting Known Good characters. * Validation should be performed on a per field basis and should endeavour to be as strict as possible. * Ensure that data is fully normalised and decoded before being compared to the filter. * All client supplied data should be HMTL encoded at the point where it is displayed to the user. This includes request data such as query string parameters and data retrieved from storage. * It is recommended that all alphanumeric characters be HTML encoded to avoid XSS. However the following characters must be encoded: double quotes, ampersand, less than sign, and greater than sign | | |
| **AFFECTED ASSETS** | | |
| **Open** - **globexcorp.com.au**  **REMEDIATION NOTES**  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  **PROOF OF CONCEPT / STEPS TO REPRODUCE**  Using a web proxy or a command line tool (such as curl or wget), submit the request attached to this finding. Notice the response contains a 'Content-Type: text/html;', and includes the HTML payload passed in the URL.  **Open** - **classic.acmegroup.co**  **REMEDIATION NOTES**  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  **PROOF OF CONCEPT / STEPS TO REPRODUCE**   1. do this. 2. do that.   <script>alert(1)</script>    **Open** - **classic.acmegroup.co**  **REMEDIATION NOTES**  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  **PROOF OF CONCEPT / STEPS TO REPRODUCE**  do this... | | |
| **EVIDENCE** | | |
| None. | | |

|  |  |  |
| --- | --- | --- |
|  | Cross Site Request Forgery (CSRF) |  |
| **CVSSv3 SCORE** | | |
| Base: **5.3**  Temporal: **5.3**  Environmental: **5.3**  Vector: **CVSS:3.1/AV:N/AC:H/PR:H/UI:R/S:U/C:L/I:L/A:H** | | |
| **DESCRIPTION** | | |
| The application was found not to implement controls against Cross-Site Request Forgery (CSRF). An attacker can exploit this issue, with some interaction with an authenticated victim, to cause a victim to perform arbitrary actions within the application. An attacker can exploit this issue by sending a malicious link to an application user, which appears to be from a legitimate application domain. When clicked, this link will cause the victims web browser to send a request of the attackers choosing to the application. | | |
| **ATTACK SCENARIO** | | |
| An attacker can exploit this issue to force legitimate users to perform arbitrary actions in the application, depending on their account privilege level. For example, an attacker can force a user to make requests to the application without the victims knowledge, however, given the limited functionality of the application it would be difficult for an attacker to force a user trigger any malicious actions. | | |
| **REMEDIATION RECOMMENDATION** | | |
| Modify the application to include a one-time token (a nonce) in each application form which performs authenticated actions. Verify that this nonce is correct for each authenticated application request, before processing this request.  Add a hash (session id, function name, server-side secret) to all forms.  For .NET, add a session identifier to ViewState with MAC (described in detail on https://www.owasp.org/index.php/Cross-Site\_Request\_Forgery\_(CSRF)\_Prevention\_Cheat\_Sheet#Viewstate\_.28ASP.NET.29).  Checking the referrer header in the client's HTTP request can prevent CSRF attacks. Ensuring that the HTTP request has come from the original site means that attacks from other sites will not function. It is very common to see referrer header checks used on embedded network hardware due to memory limitations. However XSS can be used to bypass both referrer and token based checks simultaneously.  Although CSRF is fundamentally a problem with the web application, not the user, users can help protect their accounts at poorly designed sites by logging off the site before visiting another, or clearing their browser's cookies at the end of each browser session.  See https://www.owasp.org/index.php/Cross-Site\_Request\_Forgery\_(CSRF) for more details and recommendations. | | |
| **AFFECTED ASSETS** | | |
| **Open** - **globexcorp.com.au**  **REMEDIATION NOTES**  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  **PROOF OF CONCEPT / STEPS TO REPRODUCE**  1. Open in a web browser the csrf-test.html proof-of-concept file uploaded. The form will be submitted without the users consent.  2. Open the cases tab in the app: /crm/cases  3. You will find the new cases on top of the list. | | |
| **EVIDENCE** | | |
| None. | | |

|  |  |  |
| --- | --- | --- |
|  | Functionality Misuse |  |
| **CVSSv3 SCORE** | | |
| Base: **6.5**  Temporal: **6.5**  Environmental: **6.5**  Vector: **CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:C/C:L/I:L/A:L** | | |
| **DESCRIPTION** | | |
| Abuse of functionality is an attack technique that uses a web site's own features and functionality to attack itself or others. Abuse of functionality can be described as the abuse of an application's intended functionality to perform an undesirable outcome. These attacks have varied results such as consuming resources, circumventing access controls, or leaking information. The potential and level of abuse will vary from application to application. Abuse of functionality attacks are often a combination of other attack types and/or utilise other attack vectors. | | |
| **ATTACK SCENARIO** | | |
| Outcomes can range from information exposure, vandalism, spamming, degrading or denial of service, as well as execution of arbitrary code on the target machine. Some examples of abuse of functionality include abusing send-mail functions, abusing password-recovery flows, and abusing functionality to make unrestricted proxy requests. | | |
| **REMEDIATION RECOMMENDATION** | | |
| Carefully review affected functionality and consider disabling if not required. If required, consider use case where functionality may be abused and enforce restrictions and security controls to reduce or eliminate possibility of misuse/abuse, such as server-side validation and removing input from client where avoidable; and enable monitoring and logging where appropriate. | | |
| **AFFECTED ASSETS** | | |
| **Open** - **globexcorp.com.au**  **REMEDIATION NOTES**  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  **PROOF OF CONCEPT / STEPS TO REPRODUCE**  It is possible to create a case on behalf of another user, or a non-existing-user, by modifying the 'X' parameter to another email address during request to POST /crm/test?encoding=UTF endpoint. | | |
| **EVIDENCE** | | |
| None. | | |

|  |  |  |
| --- | --- | --- |
|  | No Account Lockout May Facilitate Brute Force Password Attack |  |
| **CVSSv3 SCORE** | | |
| Base: **6.4**  Temporal: **6.4**  Environmental: **6.4**  Vector: **CVSS:3.1/AV:N/AC:H/PR:N/UI:R/S:U/C:L/I:H/A:L** | | |
| **DESCRIPTION** | | |
| The application was found to have not implemented any account lockout mechanisms. The lack of lockout functionality could allow an attacker to attempt brute force attack on differing username and password combinations in order to obtain valid application credentials. During testing a significant number of failed attempts were performed against an existing application account, followed by successful authentication. This indicates that the account was not locked out during the failed login attempts. | | |
| **ATTACK SCENARIO** | | |
| An attacker can conduct brute force password attacks to establish valid credentials for the service. This would allow authenticated access to the service potentially exposing sensitive information to an unauthorised user. | | |
| **REMEDIATION RECOMMENDATION** | | |
| It is recommended to modify the authentication functionality to implement locking of accounts after a number of unsuccessful attempts. A balance should be established to allow a reasonable number of failed attempts after which the accounts will be locked. For sensitive applications, it is recommended this value to be either three (3) or five (5). Ideally this lockout would be temporary, using an exponentially increasing duration of lockout so as to prevent brute force attacks, without introducing denial of service conditions for legitimate users of the application. | | |
| **AFFECTED ASSETS** | | |
| **Open** - **globexcorp.com.au**  **REMEDIATION NOTES**  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  **PROOF OF CONCEPT / STEPS TO REPRODUCE**  Attempt to authenticate with the wrong password more than ten (10) times in the application, then login with the correct password. Note that the application did not lock the user account although numerous attempts were performed. . | | |
| **EVIDENCE** | | |
| None. | | |

|  |  |  |
| --- | --- | --- |
|  | Open Redirection |  |
| **CVSSv3 SCORE** | | |
| Base: **6**  Temporal: **6**  Environmental: **6**  Vector: **CVSS:3.1/AV:N/AC:H/PR:L/UI:N/S:C/C:L/I:L/A:L** | | |
| **DESCRIPTION** | | |
| URL redirection functionality allows web applications seamlessly and quietly direct users to alternate locations within the application or even to allowed external resources. Open redirection occurs when this functionality can be abused to redirect users to arbitrary locations without specifically informing or advising that this redirection has occurred.  Through manipulation of location values used in the redirection facilitates an attacker to lend credibility of the application to an unrelated site controlled by the attacker. This site could then be used to perform further hostile actions against the user. The application was found to suffer from an open redirection weakness. User-controlled input that specifies a link to an external site can be injected which can be used to redirect victim application users to this location. | | |
| **ATTACK SCENARIO** | | |
| The open redirection vulnerability could be used to aid in phishing attacks against users. As the site will redirect to a URL of the attacker choice, user trust with the application can be leveraged to make a phishing attack more successful. As the URL provided in the phishing email is legitimate, a victim user may be more likely to follow the link. Once redirected the attacker can perform further hostile actions against the user.  The user may be redirected to an untrusted page that contains malware which may then compromise the user's machine. This will expose the user to extensive risk and the user's interaction with the web server may also be compromised if the malware conducts keylogging or other attacks that steal credentials, personally identifiable information (PII), or other important data. | | |
| **REMEDIATION RECOMMENDATION** | | |
| Ensure all redirect functionality is restricted and utilise a whitelist of approved URLs or domains allowed for use, when redirecting. When receiving user data, it must always be sanitised and validated by the application, prior to processing. Ensure that the data received is of expected type, length and acceptable values in line with business rules.  Assume all input is malicious. Use an 'accept known good' input validation strategy, i.e., use a whitelist of acceptable inputs that strictly conform to specifications. Reject any input that does not strictly conform to specifications, or transform it into something that does.  When performing input validation, consider all potentially relevant properties, including length, type of input, the full range of acceptable values, missing or extra inputs, syntax, consistency across related fields, and conformance to business rules. | | |
| **AFFECTED ASSETS** | | |
| **Open** - **globexcorp.com.au**  **REMEDIATION NOTES**  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  **PROOF OF CONCEPT / STEPS TO REPRODUCE**  1. Using an intercepting proxy, modify the 'retURL' parameter for the POST /crm/test?encoding=UTF-8 endpoint to another URL, for instance https://globexcorp.com.au.   2. Take note after form is processed, the user is redirected to attacker's website i.e. https://globexcorp.com.au. | | |
| **EVIDENCE** | | |
| None. | | |

|  |  |  |
| --- | --- | --- |
|  | SSL Certificate Expiry |  |
| **CVSSv3 SCORE** | | |
| Base: **5.3**  Temporal: **5.3**  Environmental: **5.3**  Vector: **CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:L/A:N** | | |
| **DESCRIPTION** | | |
| This plugin checks expiry dates of certificates associated with SSL- enabled services on the target and reports whether any have already expired.  The remote server's SSL certificate has already expired. | | |
| **ATTACK SCENARIO** | | |
| N/A | | |
| **REMEDIATION RECOMMENDATION** | | |
| Purchase or generate a new SSL certificate to replace the existing one. | | |
| **AFFECTED ASSETS** | | |
| **Open** - **classic.acmegroup.co**  **REMEDIATION NOTES**  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  **PROOF OF CONCEPT / STEPS TO REPRODUCE**  The SSL certificate has already expired :   Subject : C=XX, L=Default City, O=Default Company Ltd  Issuer : C=XX, L=Default City, O=Default Company Ltd  Not valid before : Sep 27 11:01:37 2017 GMT  Not valid after : Sep 27 11:01:37 2018 GMT | | |
| **EVIDENCE** | | |
| None. | | |

|  |  |  |
| --- | --- | --- |
|  | SSL Weak Cipher Suites Supported |  |
| **CVSSv3 SCORE** | | |
| Base: **5.3**  Temporal: **5.3**  Environmental: **5.3**  Vector: **CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N** | | |
| **DESCRIPTION** | | |
| The remote host supports the use of SSL ciphers that offer weak encryption.  Note: This is considerably easier to exploit if the attacker is on the same physical network.  The remote service supports the use of weak SSL ciphers. | | |
| **ATTACK SCENARIO** | | |
| N/A | | |
| **REMEDIATION RECOMMENDATION** | | |
| Reconfigure the affected application, if possible to avoid the use of weak ciphers. | | |
| **AFFECTED ASSETS** | | |
| **Open** - **classic.acmegroup.co**  **REMEDIATION NOTES**  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  **PROOF OF CONCEPT / STEPS TO REPRODUCE**  Here is the list of weak SSL ciphers supported by the remote server :   Low Strength Ciphers (<= 64-bit key)   EDH-RSA-DES-CBC-SHA Kx=DH Au=RSA Enc=DES-CBC(56) Mac=SHA1   DES-CBC-SHA Kx=RSA Au=RSA Enc=DES-CBC(56) Mac=SHA1   The fields above are :   {OpenSSL ciphername}  Kx={key exchange}  Au={authentication}  Enc={symmetric encryption method}  Mac={message authentication code}  {export flag} | | |
| **EVIDENCE** | | |
| None. | | |

|  |  |  |
| --- | --- | --- |
|  | SSL Certificate Signed Using Weak Hashing Algorithm |  |
| **CVSSv3 SCORE** | | |
| Base: **7.5**  Temporal: **7.5**  Environmental: **7.5**  Vector: **CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:H/A:N** | | |
| **DESCRIPTION** | | |
| The remote service uses an SSL certificate chain that has been signed using a cryptographically weak hashing algorithm (e.g. MD2, MD4, MD5, or SHA1). These signature algorithms are known to be vulnerable to collision attacks. An attacker can exploit this to generate another certificate with the same digital signature, allowing an attacker to masquerade as the affected service.  Note that this plugin reports all SSL certificate chains signed with SHA-1 that expire after January 1, 2017 as vulnerable. This is in accordance with Google's gradual sunsetting of the SHA-1 cryptographic hash algorithm.  Note that certificates in the chain that are contained in the Nessus CA database (known\_CA.inc) have been ignored.  An SSL certificate in the certificate chain has been signed using a weak hash algorithm. | | |
| **ATTACK SCENARIO** | | |
| N/A | | |
| **REMEDIATION RECOMMENDATION** | | |
| Contact the Certificate Authority to have the certificate reissued. | | |
| **AFFECTED ASSETS** | | |
| **Open** - **classic.acmegroup.co**  **REMEDIATION NOTES**  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  **PROOF OF CONCEPT / STEPS TO REPRODUCE**  The following certificates were part of the certificate chain sent by the remote host, but contain hashes that are considered to be weak.  |-Subject : C=XX/L=Default City/O=Default Company Ltd |-Signature Algorithm : SHA-1 With RSA Encryption |-Valid From : Sep 27 11:01:37 2017 GMT |-Valid To : Sep 27 11:01:37 2018 GMT | | |
| **EVIDENCE** | | |
| None. | | |

|  |  |  |
| --- | --- | --- |
|  | SSL Medium Strength Cipher Suites Supported (SWEET32) |  |
| **CVSSv3 SCORE** | | |
| Base: **7.5**  Temporal: **7.5**  Environmental: **7.5**  Vector: **CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N** | | |
| **DESCRIPTION** | | |
| The remote host supports the use of SSL ciphers that offer medium strength encryption. Nessus regards medium strength as any encryption that uses key lengths at least 64 bits and less than 112 bits, or else that uses the 3DES encryption suite.  Note that it is considerably easier to circumvent medium strength encryption if the attacker is on the same physical network.  The remote service supports the use of medium strength SSL ciphers. | | |
| **ATTACK SCENARIO** | | |
| N/A | | |
| **REMEDIATION RECOMMENDATION** | | |
| Reconfigure the affected application if possible to avoid use of medium strength ciphers. | | |
| **AFFECTED ASSETS** | | |
| **Open** - **classic.acmegroup.co**  **REMEDIATION NOTES**  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  **PROOF OF CONCEPT / STEPS TO REPRODUCE**  Medium Strength Ciphers (> 64-bit and < 112-bit key, or 3DES)   EDH-RSA-DES-CBC3-SHA Kx=DH Au=RSA Enc=3DES-CBC(168) Mac=SHA1   ECDHE-RSA-DES-CBC3-SHA Kx=ECDH Au=RSA Enc=3DES-CBC(168) Mac=SHA1   DES-CBC3-SHA Kx=RSA Au=RSA Enc=3DES-CBC(168) Mac=SHA1   The fields above are :   {OpenSSL ciphername}  Kx={key exchange}  Au={authentication}  Enc={symmetric encryption method}  Mac={message authentication code}  {export flag}  **Open** - **classic.acmegroup.co**  **REMEDIATION NOTES**  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  **PROOF OF CONCEPT / STEPS TO REPRODUCE**  Medium Strength Ciphers (> 64-bit and < 112-bit key, or 3DES)   EDH-RSA-DES-CBC3-SHA Kx=DH Au=RSA Enc=3DES-CBC(168) Mac=SHA1   ECDHE-RSA-DES-CBC3-SHA Kx=ECDH Au=RSA Enc=3DES-CBC(168) Mac=SHA1   DES-CBC3-SHA Kx=RSA Au=RSA Enc=3DES-CBC(168) Mac=SHA1   The fields above are :   {OpenSSL ciphername}  Kx={key exchange}  Au={authentication}  Enc={symmetric encryption method}  Mac={message authentication code}  {export flag} | | |
| **EVIDENCE** | | |
| None. | | |

|  |  |  |
| --- | --- | --- |
|  | SSL Certificate Cannot Be Trusted |  |
| **CVSSv3 SCORE** | | |
| Base: **6.5**  Temporal: **6.5**  Environmental: **6.5**  Vector: **CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:N** | | |
| **DESCRIPTION** | | |
| The server's X.509 certificate cannot be trusted. This situation can occur in three different ways, in which the chain of trust can be broken, as stated below :   - First, the top of the certificate chain sent by the server might not be descended from a known public certificate authority. This can occur either when the top of the chain is an unrecognized, self-signed certificate, or when intermediate certificates are missing that would connect the top of the certificate chain to a known public certificate authority.   - Second, the certificate chain may contain a certificate that is not valid at the time of the scan. This can occur either when the scan occurs before one of the certificate's 'notBefore' dates, or after one of the certificate's 'notAfter' dates.   - Third, the certificate chain may contain a signature that either didn't match the certificate's information or could not be verified. Bad signatures can be fixed by getting the certificate with the bad signature to be re-signed by its issuer. Signatures that could not be verified are the result of the certificate's issuer using a signing algorithm that Nessus either does not support or does not recognize.  If the remote host is a public host in production, any break in the chain makes it more difficult for users to verify the authenticity and identity of the web server. This could make it easier to carry out man-in-the-middle attacks against the remote host.  The SSL certificate for this service cannot be trusted. | | |
| **ATTACK SCENARIO** | | |
| N/A | | |
| **REMEDIATION RECOMMENDATION** | | |
| Purchase or generate a proper certificate for this service. | | |
| **AFFECTED ASSETS** | | |
| **Open** - **classic.acmegroup.co**  **REMEDIATION NOTES**  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  **PROOF OF CONCEPT / STEPS TO REPRODUCE**  The following certificate was part of the certificate chain sent by the remote host, but it has expired :  |-Subject : C=XX/L=Default City/O=Default Company Ltd |-Not After : Sep 27 11:01:37 2018 GMT  The following certificate was at the top of the certificate chain sent by the remote host, but it is signed by an unknown certificate authority :  |-Subject : C=XX/L=Default City/O=Default Company Ltd |-Issuer : C=XX/L=Default City/O=Default Company Ltd  **Open** - **classic.acmegroup.co**  **REMEDIATION NOTES**  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  **PROOF OF CONCEPT / STEPS TO REPRODUCE**  The following certificate was at the top of the certificate chain sent by the remote host, but it is signed by an unknown certificate authority :  |-Subject : C=au/L=admin/O=acme/CN=acme WebAdmin CA/E=dan@acmegroup.co |-Issuer : C=au/L=admin/O=acme/CN=acme WebAdmin CA/E=dan@acmegroup.co | | |
| **EVIDENCE** | | |
| None. | | |

|  |  |  |
| --- | --- | --- |
|  | SSL Self-Signed Certificate |  |
| **CVSSv3 SCORE** | | |
| Base: **NA**  Temporal: **NA**  Environmental: **NA**  Vector: **NA** | | |
| **DESCRIPTION** | | |
| The X.509 certificate chain for this service is not signed by a recognized certificate authority. If the remote host is a public host in production, this nullifies the use of SSL as anyone could establish a man-in-the-middle attack against the remote host.   Note that this plugin does not check for certificate chains that end in a certificate that is not self-signed, but is signed by an unrecognized certificate authority.  The SSL certificate chain for this service ends in an unrecognized self-signed certificate. | | |
| **ATTACK SCENARIO** | | |
| N/A | | |
| **REMEDIATION RECOMMENDATION** | | |
| Purchase or generate a proper certificate for this service. | | |
| **AFFECTED ASSETS** | | |
| **Open** - **classic.acmegroup.co**  **REMEDIATION NOTES**  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  **PROOF OF CONCEPT / STEPS TO REPRODUCE**  The following certificate was found at the top of the certificate chain sent by the remote host, but is self-signed and was not found in the list of known certificate authorities :  |-Subject : C=XX/L=Default City/O=Default Company Ltd  **Open** - **classic.acmegroup.co**  **REMEDIATION NOTES**  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  **PROOF OF CONCEPT / STEPS TO REPRODUCE**  The following certificate was found at the top of the certificate chain sent by the remote host, but is self-signed and was not found in the list of known certificate authorities :  |-Subject : C=au/L=admin/O=acme/CN=acme WebAdmin CA/E=dan@acmegroup.co | | |
| **EVIDENCE** | | |
| None. | | |

|  |  |  |
| --- | --- | --- |
|  | Multiple Simultaneous User Sessions |  |
| **CVSSv3 SCORE** | | |
| Base: **3.7**  Temporal: **3.7**  Environmental: **3.7**  Vector: **CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N** | | |
| **DESCRIPTION** | | |
| It was identified that the application does not limit a number of simultaneous logins with a single user account. A user account can be used to log in from multiple locations or browsers at the same time, without the real authenticated user being notified of such events. While technically not considered a vulnerability, this issue can be leveraged by an attacker to maintain access to a compromised account with reduced risk of being detected. | | |
| **ATTACK SCENARIO** | | |
| An attacker with access to application through a compromised account is able to continue using this access with a lowered risk of detection. The weakness is simple to utilise and is by application design. However an attacker would require a compromised user account in order to leverage this issue. | | |
| **REMEDIATION RECOMMENDATION** | | |
| It is recommended to restrict or constrain application users to a single logon session only. Implementation of this restriction can be achieved by destroying any prior sessions upon authentication, or by denying subsequent authentication should a valid and active session already exist with the server.  Additionally, consider implementing functionality that upon a user successfully authenticating to the application, they are are shown the date, time, and IP address of the last login of the user. | | |
| **AFFECTED ASSETS** | | |
| **Open** - **globexcorp.com.au**  **REMEDIATION NOTES**  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  **PROOF OF CONCEPT / STEPS TO REPRODUCE**  Access the application in multiple browsers and take note that simultaneous sessions are permitted. Also take note that no login information for the previous session is displayed to the user on the login landing page. | | |
| **EVIDENCE** | | |
| None. | | |

|  |  |  |
| --- | --- | --- |
|  | HTML 5 Cross Origin Resource Sharing |  |
| **CVSSv3 SCORE** | | |
| Base: **3.7**  Temporal: **3.7**  Environmental: **3.7**  Vector: **CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N** | | |
| **DESCRIPTION** | | |
| The HTML5 cross-origin resource sharing policy controls whether and how content running on other domains can perform two-way interaction with the domain which publishes the policy. | | |
| **ATTACK SCENARIO** | | |
| If another domain is allowed by the policy, then that domain can potentially attack users of the application. If a user is logged in to the application, and visits a domain allowed by the policy, then any malicious content running on that domain can potentially retrieve content from the application, and sometimes carry out actions within the security context of the logged in user.  Even if an allowed domain is not overtly malicious in itself, security vulnerabilities within that domain could potentially be leveraged by a third-party attacker to exploit the trust relationship and attack the application which allows access. | | |
| **REMEDIATION RECOMMENDATION** | | |
| It is recommended to review the domains which are allowed by the CORS policy in relation to any sensitive content within the application, and determine whether it is appropriate for the application to trust both the intentions and security posture of those domains. | | |
| **AFFECTED ASSETS** | | |
| **Open** - **globexcorp.com.au**  **REMEDIATION NOTES**  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  *01/06/2024*  Issue Re-Opened: Issue has not been fixed  **PROOF OF CONCEPT / STEPS TO REPRODUCE**  Access the application with a browser and an intercepting proxy. Take note that the application allows cross origin resource sharing for all origins, specified by the asterisk on API calls. | | |
| **EVIDENCE** | | |
| None. | | |

|  |  |  |
| --- | --- | --- |
|  | Missing X-XSS-Protection Header |  |
| **CVSSv3 SCORE** | | |
| Base: **3.7**  Temporal: **3.7**  Environmental: **3.7**  Vector: **CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N** | | |
| **DESCRIPTION** | | |
| The X-XSS-Protection HTTP Header is not set. This header enables the Cross-Site Scripting (XSS) filter built into most recent web browsers. By setting the X-XSS-Protection header the browser is instructed to not render the page if an XSS attack is detected. It is therefore a client-side defence mechanism. | | |
| **ATTACK SCENARIO** | | |
| Disabling (not using) this option increases the attack surface for XSS attacks. | | |
| **REMEDIATION RECOMMENDATION** | | |
| Add the following HTTP header to all server responses: X-XSS-Protection: 1; mode=block. For more information visit https://www.owasp.org/index.php/OWASP\_Secure\_Headers\_Project | | |
| **AFFECTED ASSETS** | | |
| **Open** - **globexcorp.com.au**  **REMEDIATION NOTES**  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  **PROOF OF CONCEPT / STEPS TO REPRODUCE**  Access the application with a browser and an intercepting proxy. Take note that the application does not set the X-XSS-Protection header in HTTP responses. | | |
| **EVIDENCE** | | |
| None. | | |

|  |  |  |
| --- | --- | --- |
|  | Cookie Without HTTPOnly Flag Set |  |
| **CVSSv3 SCORE** | | |
| Base: **3.7**  Temporal: **3.7**  Environmental: **3.7**  Vector: **CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:N/I:L/A:N** | | |
| **DESCRIPTION** | | |
| The HttpOnly security feature restricts JavaScript from accessing cookie values. This feature was implemented to protect cookie values from being obtained through cross-site scripting attacks, with the intent to deny disclosure of session tokens. While the attribute, in those browsers which support it, provides additional protection for these sensitive items, HttpOnly is not considered a complete solution. It is noted that functionality such as XMLHTTPRequest and similar browser technologies, provide read access to HTTP headers including the Set-Cookie header. It is in this header that the HttpOnly feature is set. | | |
| **ATTACK SCENARIO** | | |
| An attacker that has identified a cross-site scripting weakness in the application has the ability to access the unprotected cookie values more easily. Using these values, further attacks or access could gained. In this instance the anti-CSRF key could be accessed. | | |
| **REMEDIATION RECOMMENDATION** | | |
| It is considered best practice to apply layered defensive measures to protect application data. As such, it is recommended to apply the HttpOnly attribute to all cookie values set by the application. It is noted that in some instances, that application design or business requirements require scripts to access cookie values and as such, the HttpOnly attribute cannot be applied without adverse effects on functionality. These design restrictions should however be reviewed and avoided where possible.  For more details about the HttpOnly attribute see the following reference:  - Introduction to HttpOnly https://www.owasp.org/index.php/HttpOnly | | |
| **AFFECTED ASSETS** | | |
| **Open** - **globexcorp.com.au**  **REMEDIATION NOTES**  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  **PROOF OF CONCEPT / STEPS TO REPRODUCE**  To reproduce this issue configure an intercepting proxy and monitor application traffic. Note the absence of HTTPOnly flag in set-cookie header. | | |
| **EVIDENCE** | | |
| None. | | |

|  |  |  |
| --- | --- | --- |
|  | Strict Transport Security Policy Not Enforced |  |
| **CVSSv3 SCORE** | | |
| Base: **3.7**  Temporal: **3.7**  Environmental: **3.7**  Vector: **CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:N/I:L/A:N** | | |
| **DESCRIPTION** | | |
| The HTTP Strict Transport Security (HSTS) policy forces the client browser to send all web requests to the server using HTTPS. When the HSTS policy is enforced, through use of appropriate HTTP headers, any attempts to connect to the server using HTTP will be automatically forced over the encrypted HTTPS channel by the browser. Furthermore, the HSTS policy ensures the user is unable to interact with the application should the servers transport layer security (TLS) certificate be unable to be trusted.  The implementation of this security header will ensure users of the application have additional protection against man in the middle type attacks, which could force requests over HTTP to disclose sensitive information through use of attacks, such as SSL Stripping. | | |
| **ATTACK SCENARIO** | | |
| Without an enforced HSTS policy, an attacker can more easily conduct man-in-the-middle attacks against application users and potentially view sensitive information, including session cookies, without the user being aware that the integrity of their encrypted session has been compromised. | | |
| **REMEDIATION RECOMMENDATION** | | |
| It is recommended to enforce the Strict-Transport-Security policy by applying the following header with all served web requests: Strict-Transport-Security: max-age=31536000; includeSubDomains  For more information about the HSTS policy see: https://www.owasp.org/index.php/HTTP\_Strict\_Transport\_Security | | |
| **AFFECTED ASSETS** | | |
| **Open** - **globexcorp.com.au**  **REMEDIATION NOTES**  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  *01/06/2024*  Issue Re-Opened: Issue has not been fixed  **PROOF OF CONCEPT / STEPS TO REPRODUCE**  This vulnerability can be observed from the HTTP response of the application server, note HSTS header is missing. This security feature can be enabled by editing the web.config file. | | |
| **EVIDENCE** | | |
| None. | | |

|  |  |  |
| --- | --- | --- |
|  | Outdated Version of JQuery |  |
| **CVSSv3 SCORE** | | |
| Base: **3.1**  Temporal: **3.1**  Environmental: **3.1**  Vector: **CVSS:3.1/AV:N/AC:H/PR:L/UI:N/S:U/C:N/I:L/A:N** | | |
| **DESCRIPTION** | | |
| It was observed during testing that the application is running on an outdated version of JQuery. | | |
| **ATTACK SCENARIO** | | |
| Security risks are a primary danger of older technology. The older the library is, the longer time hackers have to find exploitable vulnerabilities. It is even more critical in a situation when the manufacturer of software is no longer maintaining support for old/legacy libraries. | | |
| **REMEDIATION RECOMMENDATION** | | |
| Update to the latest stable version of JQuery. | | |
| **AFFECTED ASSETS** | | |
| **Open** - **globexcorp.com.au**  **REMEDIATION NOTES**  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  **PROOF OF CONCEPT / STEPS TO REPRODUCE**  Access any of the listed files below and note the jQuery version mentioned on top of the file. - /crm/test/file1.html - /crm/test/file2.html - /crm/test/file3.html | | |
| **EVIDENCE** | | |
| None. | | |

|  |  |  |
| --- | --- | --- |
|  | Cookie With Secure Flag Missing |  |
| **CVSSv3 SCORE** | | |
| Base: **3.7**  Temporal: **3.7**  Environmental: **3.7**  Vector: **CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:N/I:L/A:N** | | |
| **DESCRIPTION** | | |
| If the secure flag is set on a cookie, then browsers will not submit the cookie in any requests that use an unencrypted HTTP connection, thereby preventing the cookie from being trivially intercepted by an attacker monitoring network traffic.  If the secure flag is not set, then the cookie will be transmitted in clear-text if the user visits any HTTP URLs within the cookie's scope. An attacker may be able to induce this event by feeding a user suitable links, either directly or via another web site. Even if the domain which issued the cookie does not host any content that is accessed over HTTP, an attacker may be able to use links of the form http://example.com:443/ to perform the same attack.   Secure flag on cookies is often either misconfigured or overlooked during application development and subsequently transferred to the production environment. | | |
| **ATTACK SCENARIO** | | |
| An attacker may be able to force the users browser to make a request through the use of other application or server weaknesses, over an insecure channel. As the cookie is not protected by the secure attribute, the browser will honour this request, passing the unprotected session cookie. The cookie value may then be disclosed to other parties when passed over the insecure channel.   This issue cannot be exploited alone an attacker must exploit a secondary attack vector, such as a man-in-the-middle attack or cross-site scripting to exploit this issue.   This issue alone provides minimal benefit to an attacker. However, this makes it easier for an attacker to exploit issues such as cross-site scripting. | | |
| **REMEDIATION RECOMMENDATION** | | |
| The secure flag should be set on all cookies that are used for transmitting sensitive data when accessing content over HTTPS. Session tokens in use should never be transmitted over unencrypted communications. | | |
| **AFFECTED ASSETS** | | |
| **Open** - **globexcorp.com.au**  **REMEDIATION NOTES**  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  **PROOF OF CONCEPT / STEPS TO REPRODUCE**  To reproduce this issue configure an intercepting proxy and monitor application traffic. Note the absence of secure flag in set-cookie header. | | |
| **EVIDENCE** | | |
| None. | | |

|  |  |  |
| --- | --- | --- |
|  | User Enumeration |  |
| **CVSSv3 SCORE** | | |
| Base: **3.7**  Temporal: **3.7**  Environmental: **3.7**  Vector: **CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N** | | |
| **DESCRIPTION** | | |
| The application responds differently depending on whether a valid or invalid username is supplied by the user. This allows an attacker to enumerate valid application usernames prior to a brute-force password guessing attack, which greatly increases the chances of an attacker gaining unauthorised access to the application. The attacker will generate a large list of different emails, submit these emails one by one to the forms and identify emails that result in positive application response. Since both a valid email and password are required for successful authentication, knowledge of a valid user name is critical for a consequent guessing attack against user password. Enumerated emails also can be used for targeted spam campaigns or phishing attacks (also called spear fishing attack). | | |
| **ATTACK SCENARIO** | | |
| This issue requires basic technical knowledge to exploit, and can be easily exploited by measuring the length of the response or looking for certain keywords that change if an account exists or not. The attack is trivial. In its basic form the attack does not require any special tools or techniques however in order to enumerate a large number of usernames automated tools are a better and more effective option. There are a number of free tools that offer fast enumeration of thousands of username records. Although email is often not considered as sensitive information it is an essential component of authentication process without which a successful password brute force attack would not be possible. | | |
| **REMEDIATION RECOMMENDATION** | | |
| Modify functionality to return only a generic response making it impossible to distinguish between a valid username and an invalid username. | | |
| **AFFECTED ASSETS** | | |
| **Open** - **globexcorp.com.au**  **REMEDIATION NOTES**  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  *01/06/2024*  Issue Re-Opened: Issue has not been fixed  **PROOF OF CONCEPT / STEPS TO REPRODUCE**  Try to sign up for a new user. Supply the sign up form with an email that belongs to an existing account, and notice the application displays a different error message saying an account already exists with the email. | | |
| **EVIDENCE** | | |
| None. | | |

|  |  |  |
| --- | --- | --- |
|  | Server Discloses Supporting Technology |  |
| **CVSSv3 SCORE** | | |
| Base: **3.7**  Temporal: **3.7**  Environmental: **3.7**  Vector: **CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N** | | |
| **DESCRIPTION** | | |
| The application was found to disclose supporting technology used by the server. Information disclosure is a common and prevalent vulnerability in web applications. The disclosure of sensitive information either directly or implicitly through application behaviour, may aid an attacker with information gathering or profiling, and determining or establishing other vectors of attack against the application or host. | | |
| **ATTACK SCENARIO** | | |
| Whilst information disclosure is prevalent and easily obtained, in order to exploit this an attacker would need to leverage this information to find publically known vulnerabilities or exploits for the identified server, or use this information to tailor specific exploits. Using the information obtained from the HTTP header values and those disclosed by the default messages, the attacker would then conduct research to ascertain if any publically known vulnerabilities or exploits exist for the identified server, or use this information to tailor specific exploits. | | |
| **REMEDIATION RECOMMENDATION** | | |
| Configure the web server config to not return this supporting technology information in headers. | | |
| **AFFECTED ASSETS** | | |
| **Open** - **globexcorp.com.au**  **REMEDIATION NOTES**  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  **PROOF OF CONCEPT / STEPS TO REPRODUCE**  Access the application with a browser and an intercepting proxy. Take note that the application discloses the software and version of nginx in use in HTTP responses. | | |
| **EVIDENCE** | | |
| None. | | |

|  |  |  |
| --- | --- | --- |
|  | Insufficient Framing Protection |  |
| **CVSSv3 SCORE** | | |
| Base: **3.7**  Temporal: **3.7**  Environmental: **3.7**  Vector: **CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N** | | |
| **DESCRIPTION** | | |
| The act of placing one web page inside another web page is referred to as framing. This is performed through the use of HTML tag called an iframe. An example where framing is often performed is for the inclusion of content from other sources such as embedding of maps or videos. Should a trusted application be framed within an untrusted web page, click-jacking or frame sniffing attacks against application user can be performed. The application was found to be lacking appropriate X-Frame-Options headers which significantly reduces likelihood against click-jacking and frame sniffing attacks. | | |
| **ATTACK SCENARIO** | | |
| An attacker may be able to get an authenticated user to inadvertently perform actions by luring or enticing them to access a maliciously crafted web page. By clicking on the page, actions could be performed on the users behalf, through the framed page. There are few examples available on the Internet of how to build a clickjacking web page, however using such examples in a real life attack would most likely require significant customisation in order to take advantage of the particular functions within the Web front end for the application. | | |
| **REMEDIATION RECOMMENDATION** | | |
| To prevent framing attacks, include the X-Frame-Options HTTP header for all web application page responses.  The specified values available for X-Frame-Options are:  - DENY The page cannot be displayed in a frame  - SAMEORIGIN The page can only be displayed in a frame on the same origin as the page itself.  - ALLOW-FROM URI The page can only be displayed in a frame on the specified origin   Typically the DENY option is suitable for most web applications. | | |
| **AFFECTED ASSETS** | | |
| **Open** - **globexcorp.com.au**  **REMEDIATION NOTES**  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  **PROOF OF CONCEPT / STEPS TO REPRODUCE**  Access the application with an intercepting proxy and a browser. Take note that the application does not set X-Frame-Options headers in HTTP responses. | | |
| **EVIDENCE** | | |
| None. | | |

|  |  |  |
| --- | --- | --- |
|  | Insufficient Session Expiration |  |
| **CVSSv3 SCORE** | | |
| Base: **3.7**  Temporal: **3.7**  Environmental: **3.7**  Vector: **CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N** | | |
| **DESCRIPTION** | | |
| It was discovered the application maintains users session after clicking the logout link, or after a period of inactivity (30 minutes or more).  If browser back button is clicked after logging out, or a captured authenticated request is resent to the server, the application/server will honour this request indicating session has not been properly invalidated. | | |
| **ATTACK SCENARIO** | | |
| An attacker can resume a session that has not been properly invalidated and access the functionality available to the user of that session. This attack or exposure can be more damaging and practical in the case where shared or public computers are used. | | |
| **REMEDIATION RECOMMENDATION** | | |
| It is recommended that session expiration mechanisms are employed ensuring user sessions are effectively expired and become unusable once the logout request has been issued, and once the token time-to-live (TTL) has expired. | | |
| **AFFECTED ASSETS** | | |
| **Open** - **globexcorp.com.au**  **REMEDIATION NOTES**  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  **PROOF OF CONCEPT / STEPS TO REPRODUCE**  Using an intercepting proxy, replay API calls to the application after a user has logged out. Take note that the session token remains active. | | |
| **EVIDENCE** | | |
| None. | | |

|  |  |  |
| --- | --- | --- |
|  | SSL RC4 Cipher Suites Supported (Bar Mitzvah) |  |
| **CVSSv3 SCORE** | | |
| Base: **NA**  Temporal: **NA**  Environmental: **NA**  Vector: **NA** | | |
| **DESCRIPTION** | | |
| The remote host supports the use of RC4 in one or more cipher suites. The RC4 cipher is flawed in its generation of a pseudo-random stream of bytes so that a wide variety of small biases are introduced into the stream, decreasing its randomness.  If plaintext is repeatedly encrypted (e.g., HTTP cookies), and an attacker is able to obtain many (i.e., tens of millions) ciphertexts, the attacker may be able to derive the plaintext.  The remote service supports the use of the RC4 cipher. | | |
| **ATTACK SCENARIO** | | |
| N/A | | |
| **REMEDIATION RECOMMENDATION** | | |
| Reconfigure the affected application, if possible, to avoid use of RC4 ciphers. Consider using TLS 1.2 with AES-GCM suites subject to browser and web server support. | | |
| **AFFECTED ASSETS** | | |
| **Open** - **form.acmegroup.co**  **REMEDIATION NOTES**  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  **PROOF OF CONCEPT / STEPS TO REPRODUCE**  List of RC4 cipher suites supported by the remote server :   High Strength Ciphers (>= 112-bit key)   ECDHE-RSA-RC4-SHA Kx=ECDH Au=RSA Enc=RC4(128) Mac=SHA1   RC4-MD5 Kx=RSA Au=RSA Enc=RC4(128) Mac=MD5   RC4-SHA Kx=RSA Au=RSA Enc=RC4(128) Mac=SHA1   The fields above are :   {OpenSSL ciphername}  Kx={key exchange}  Au={authentication}  Enc={symmetric encryption method}  Mac={message authentication code}  {export flag} | | |
| **EVIDENCE** | | |
| None. | | |

|  |  |  |
| --- | --- | --- |
|  | Inducing Account Lockout |  |
| **CVSSv3 SCORE** | | |
| Base: **0**  Temporal: **0**  Environmental: **0**  Vector: **CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:N/I:N/A:N** | | |
| **DESCRIPTION** | | |
| An attacker leverages the security functionality of the system aimed at thwarting potential attacks to launch a denial of service attack against a legitimate system user. Many systems, for instance, implement a password throttling mechanism that locks an account after a certain number of incorrect log in attempts. An attacker can leverage this throttling mechanism to lock a legitimate user out of their own account. The weakness that is being leveraged by an attacker is the very security feature that has been put in place to counteract attacks. | | |
| **ATTACK SCENARIO** | | |
| A famous example of this type an attack is the eBay attack. eBay always displays the user id of the highest bidder. In the final minutes of the auction, one of the bidders could try to log in as the highest bidder three times. After three incorrect log in attempts, eBay password throttling would kick in and lock out the highest bidder's account for some time. An attacker could then make their own bid and their victim would not have a chance to place the counter bid because they would be locked out. Thus an attacker could win the auction. | | |
| **REMEDIATION RECOMMENDATION** | | |
| Implement intelligent password throttling mechanisms such as those which take IP address into account, in addition to the login name. When implementing security features, consider how they can be misused and made to turn on themselves. | | |
| **AFFECTED ASSETS** | | |
| **Open** - **globexcorp.com.au**  **REMEDIATION NOTES**  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  **PROOF OF CONCEPT / STEPS TO REPRODUCE**  1. do this..  2. do that.. | | |
| **EVIDENCE** | | |
| None. | | |

|  |  |  |
| --- | --- | --- |
|  | TLS Version 1.1 Protocol Detection |  |
| **CVSSv3 SCORE** | | |
| Base: **NA**  Temporal: **NA**  Environmental: **NA**  Vector: **NA** | | |
| **DESCRIPTION** | | |
| The remote service accepts connections encrypted using TLS 1.1. TLS 1.1 lacks support for current and recommended cipher suites. Ciphers that support encryption before MAC computation, and authenticated encryption modes such as GCM cannot be used with TLS 1.1  PCI DSS v3.2 still allows TLS 1.1 as of June 30, 2018, but strongly recommends the use of TLS 1.2. A proposal is currently before the IETF to fully deprecate TLS 1.1 and many vendors have already proactively done this.  The remote service encrypts traffic using an older version of TLS. | | |
| **ATTACK SCENARIO** | | |
| N/A | | |
| **REMEDIATION RECOMMENDATION** | | |
| Enable support for TLS 1.2 and/or 1.3, and disable support for TLS 1.1. | | |
| **AFFECTED ASSETS** | | |
| **Open** - **classic.acmegroup.co**  **REMEDIATION NOTES**  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  **PROOF OF CONCEPT / STEPS TO REPRODUCE**  TLSv1.1 is enabled and the server supports at least one cipher. | | |
| **EVIDENCE** | | |
| None. | | |

|  |  |  |
| --- | --- | --- |
|  | nginx HTTP Server Detection |  |
| **CVSSv3 SCORE** | | |
| Base: **NA**  Temporal: **NA**  Environmental: **NA**  Vector: **NA** | | |
| **DESCRIPTION** | | |
| Nessus was able to detect the nginx HTTP server by looking at the HTTP banner on the remote host.  The nginx HTTP server was detected on the remote host. | | |
| **ATTACK SCENARIO** | | |
| N/A | | |
| **REMEDIATION RECOMMENDATION** | | |
| n/a | | |
| **AFFECTED ASSETS** | | |
| **Open** - **classic.acmegroup.co**  **REMEDIATION NOTES**  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  **PROOF OF CONCEPT / STEPS TO REPRODUCE**  URL : https://classic.acmegroup.co/  Version : 1.10.2  source : Server: nginx/1.10.2 | | |
| **EVIDENCE** | | |
| None. | | |

|  |  |  |
| --- | --- | --- |
|  | TLS Version 1.0 Protocol Detection |  |
| **CVSSv3 SCORE** | | |
| Base: **NA**  Temporal: **NA**  Environmental: **NA**  Vector: **NA** | | |
| **DESCRIPTION** | | |
| The remote service accepts connections encrypted using TLS 1.0. TLS 1.0 has a number of cryptographic design flaws. Modern implementations of TLS 1.0 mitigate these problems, but newer versions of TLS like 1.1 and 1.2 are designed against these flaws and should be used whenever possible.  PCI DSS v3.2 requires that TLS 1.0 be disabled entirely by June 30, 2018, except for POS POI terminals (and the SSL/TLS termination points to which they connect) that can be verified as not being susceptible to any known exploits.  The remote service encrypts traffic using an older version of TLS. | | |
| **ATTACK SCENARIO** | | |
| N/A | | |
| **REMEDIATION RECOMMENDATION** | | |
| Enable support for TLS 1.1 and 1.2, and disable support for TLS 1.0. | | |
| **AFFECTED ASSETS** | | |
| **Open** - **classic.acmegroup.co**  **REMEDIATION NOTES**  *06/20/2024*  Issue Re-Opened: Issue has not been fixed  *06/20/2024*  Issue ready for retesting  **PROOF OF CONCEPT / STEPS TO REPRODUCE**  TLSv1 is enabled and the server supports at least one cipher. | | |
| **EVIDENCE** | | |
| None. | | |

# **TEST CASES**

|  |
| --- |
| COMPLETED |

1. **Verify that the forgotten password function and other recovery paths do not reveal the current password and that the new password is not sent in clear text to the user.**  – AttackForge Admin on Saturday, January 6, 2024
2. **Verify all authentication controls fail securely to ensure attackers cannot log in.** – AttackForge Admin on Saturday, April 15, 2023
3. **Verify that input validation routines are enforced on the server side.**  – AttackForge Admin on Saturday, January 6, 2024
4. **Verify that on the server, all cached ortemporary copies ofsensitive data stored are protected from unauthorized access or purged/invalidated after the authorized user accesses the sensitive data.** – AttackForge Admin on Thursday, May 11, 2023
5. **Verify the application code does not execute uploaded data obtained from untrusted sources.** – AttackForge Admin on Thursday, May 11, 2023
6. **Verify that the Content Security Policy V2 (CSP) is in use for sites where content should not be viewed in a 3rd-party X-Frame.**  – AttackForge Admin on Thursday, May 11, 2023
7. **Verify that access controls fail securely.**  – AttackForge Admin on Saturday, January 6, 2024
8. **Verify that the application is not susceptible to OS Command Injection, or that security controls prevent OS Command Injection.**  – AttackForge Admin on Thursday, May 11, 2023
9. **Verify that session ids are sufficiently long, random and unique across the correct active session base.**  – AttackForge Admin on Saturday, January 6, 2024
10. **Verify that all user and data attributes and policy information used by access controls cannot be manipulated by end users unless specifically authorized.**  – AttackForge Admin on Saturday, January 6, 2024
11. **Make sure untrusted HTML from WYSIWYG editors or similar are properly sanitized with an HTML sanitizer and handle it appropriately according to the input validation task and encoding task.** – AttackForge Admin on Thursday, May 11, 2023
12. **Verify that all pages that require authentication have easy and visible access to logout functionality.**  – AttackForge Admin on Saturday, January 6, 2024
13. **Verify there are no default passwords in use for the application framework or any components used by the application (such as “admin/password”).**  – AttackForge Admin on Saturday, April 15, 2023
14. **Verify that all password fields do not echo the user’s password when it is entered.** – AttackForge Admin on Saturday, January 6, 2024
15. **Verify that every HTTP response contains a content type header specifying a safe character set (e.g., UTF-8, ISO 8859-1).**  – AttackForge Admin on Thursday, May 11, 2023
16. **Verify that server side input validation failures result in request rejection and are logged.**  – AttackForge Admin on Saturday, January 6, 2024
17. **Verify that security logs are protected from unauthorized access and modification.**  – AttackForge Admin on Thursday, May 11, 2023
18. **Verify that the Content Security Policy V2 (CSP) isin use in a way that either disables inline JavaScript or provides an integrity check on inline JavaScript with CSP noncing or hashing.** – AttackForge Admin on Thursday, May 11, 2023
19. **Do not use Flash, Active-X, Silverlight,NACL, client-side Java or other clientside technologies not supported natively via W3C browser standards.**  – AttackForge Admin on Thursday, May 11, 2023
20. **Verify that all successful authentication and re-authentication generates a new session and session id.**  – AttackForge Admin on Saturday, January 6, 2024
21. **Ensure that all string variables placed into HTML or other web client code is either properly contextually encoded manually, or utilise templates that automatically encode contextually to ensure the application is not susceptible to reflected, stored and DOM Cross-Site Scripting (XSS) attacks.** – AttackForge Admin on Thursday, May 11, 2023
22. **Verify that measures are in place to block the use of commonly chosen passwords and weak passphrases.** – AttackForge Admin on Saturday, January 6, 2024
23. **Verify the application will only process business logic flows in sequentialstep order, with all steps being processed in realistic human time, and not process out of order, skipped steps, process steps from another user, or too quickly submitted transactions.** – AttackForge Admin on Thursday, May 11, 2023
24. **Verify re-authentication, step up or adaptive authentication, two factor authentication, or transaction signing is required before any application-specific sensitive operations are permitted as per the risk profile of the application.** – AttackForge Admin on Saturday, January 6, 2024
25. **Verify that all suspicious authentication decisions are logged. This should include requests with relevant metadata needed for security investigations.**  – AttackForge Admin on Saturday, January 6, 2024
26. **Verify password entry fields allow, or encourage, the use of passphrases, and do not prevent long passphrases/highly complex passwords being entered.** – Test User137 on Saturday, April 15, 2023
27. **Verify that if knowledge based questions (also known as 'secret questions') are required, the questions should be strong enough to protect the application.** – AttackForge Admin on Saturday, April 15, 2023
28. **Verify security logging controls provide the ability to log success and particularly failure events that are identified as security-relevant.**  – AttackForge Admin on Thursday, May 11, 2023
29. **Verify that authenticated data is cleared from client storage, such as the browser DOM, after the session is terminated.**  – AttackForge Admin on Thursday, May 11, 2023
30. **Verify that the application has defenses against HTTP parameter pollution attacks, particularly if the application framework makes no distinction about the source of request parameters (GET, POST, cookies, headers, environment, etc.)** – AttackForge Admin on Thursday, May 11, 2023
31. **Verify that unstructured data is sanitised to enforce generic safety measures such as allowed characters and length, and characters potentially harmful in given context should be escaped (e.g. natural titles with Unicode or apostrophes).** – AttackForge Admin on Thursday, May 11, 2023
32. **Verify that account passwords make use of a sufficient strength encryption routine and that it withstands brute force attack against the encryption routine.**  – AttackForge Admin on Saturday, April 15, 2023
33. **Verify that the session id is never disclosed in URLs, error messages, or logs. This includes verifying that the application does not support URL rewriting of session cookies.** – AttackForge Admin on Saturday, January 6, 2024
34. **Verify that XML or JSON schema is in place and verified before accepting input.**  – AttackForge Admin on Thursday, May 11, 2023
35. **Verify that directory browsing/indexing is disabled unless deliberately desired. Additionally, applications should not allow discovery or disclosure of file or directory metadata, such as Thumbs.db, .DS\_Store, .git or .svn folders.** – AttackForge Admin on Saturday, January 6, 2024
36. **Verify application is not running on an outdated version of web server.** – AttackForge Admin on Thursday, May 11, 2023
37. **Verify \_\_VIEWSTATE parameter is encrypted.** – AttackForge Admin on Thursday, May 11, 2023
38. **Verify when parsing JSON in browsers, that JSON.parse is used to parse JSONon the client. Do not use eval() to parse JSON on the client.**  – AttackForge Admin on Thursday, May 11, 2023
39. **Verify that administrative interfaces are not accessible to untrusted parties.** – AttackForge Admin on Saturday, January 6, 2024
40. **Verify that proper certification revocation, such as Online Certificate Status Protocol (OSCP) Stapling, is enabled and configured.**  – AttackForge Admin on Thursday, May 11, 2023
41. **Verify that only strong algorithms, ciphers, and protocols are used, through all the certificate hierarchy, including root and intermediary certificates of your selected certifying authority. This includes verifying weak RC4 cipher or cipher less than 128bits is not in use.** – AttackForge Admin on Thursday, May 11, 2023
42. **Verify that an audit log or similar allows for non-repudiation of key transactions.**  – AttackForge Admin on Thursday, May 11, 2023
43. **Verify the use of session-based authentication and authorisation.** – AttackForge Admin on Thursday, May 11, 2023
44. **Verify that HTTP headers added by a trusted proxy or SSO devices, such as a bearer token, are authenticated by the application.**  – AttackForge Admin on Thursday, May 11, 2023
45. **Verify that if an application allows usersto authenticate,they can authenticate using two-factor authentication or other strong authentication, or any similarscheme that provides protection against usertitle + password disclosure.** – AttackForge Admin on Saturday, January 6, 2024
46. **Verify that TLS is used for all connections (including both external and backend connections) that are authenticated or that involve sensitive data or functions, and does not fall back to insecure or unencrypted protocols. Ensure the strongest alternative is the preferred algorithm.** – AttackForge Admin on Thursday, May 11, 2023
47. **Verify that all sensitive data is sent to the server in the HTTP message body or headers (i.e., URL parameters are never used to send sensitive data).**  – AttackForge Admin on Thursday, May 11, 2023
48. **Verify thatthe application build and deployment processes are performed in a secure fashion.**  – AttackForge Admin on Thursday, May 11, 2023
49. **Verify that the application is not susceptible to common XML attacks, such as XPath query tampering, XML External Entity attacks, and XML injection attacks.** – AttackForge Admin on Thursday, May 11, 2023
50. **Verify that all forms containing sensitive information have disabled client side caching, including autocomplete features.** – AttackForge Admin on Thursday, May 11, 2023
51. **Verify accessing sensitive data is logged, if the data is collected under relevant data protection directives or where logging of accesses is required.**  – AttackForge Admin on Thursday, May 11, 2023
52. **Verify that cryptographic algorithms used by the application have been validated against FIPS 140-2 or an equivalent standard.**  – AttackForge Admin on Thursday, May 11, 2023
53. **Verify application sets appropriate X-Frame-Options header for all application responses, such as DENY option.** – AttackForge Admin on Thursday, May 11, 2023
54. **Verify the system can protect against aggregate or continuous access of secured functions, resources, or data. For example, consider the use of a resource governor to limit the number of edits per hour or to prevent the entire database from being scraped by an individual user.** – AttackForge Admin on Saturday, January 6, 2024
55. **Verify that all input data is validated, not only HTML form fields but all sources of input such as REST calls, query parameters, HTTP headers, cookies, batch files, RSS feeds, etc; using positive validation (whitelisting), then lesser forms of validation such as greylisting (eliminating known bad strings), or rejecting bad inputs (blacklisting).** – AttackForge Admin on Thursday, May 11, 2023
56. **Verify that the principle of least privilege exists - users should only be able to access functions, data files, URLs, controllers, services, and other resources, for which they possess specific authorisation. This implies protection against spoofing and elevation of privilege.** – AttackForge Admin on Saturday, January 6, 2024
57. **Verify that the application correctly enforces context-sensitive authorisation so asto not allow unauthorised manipulation by means of parameter tampering.**  – AttackForge Admin on Saturday, January 6, 2024
58. **Verify that a path can be built from a trusted CA to each Transport Layer Security (TLS) server certificate, and that each server certificate is valid.**  – AttackForge Admin on Thursday, May 11, 2023
59. **Communications between components, such as between the application server and the database server, should be encrypted, particularly when the components are in different containers or on different systems.** – AttackForge Admin on Thursday, May 11, 2023
60. **Verify that files obtained from untrusted sources are validated to be of expected type and scanned by antivirus scanners to prevent upload of known malicious content.** – AttackForge Admin on Thursday, May 11, 2023
61. **If the application framework allows automatic mass parameter assignment (also called automatic variable binding) from the inbound request to a model, verify that security sensitive fields such as “accountBalance”, “role” or “password” are protected from malicious automatic binding.** – AttackForge Admin on Thursday, May 11, 2023
62. **Verify that all input is limited to an appropriate size limit.**  – AttackForge Admin on Thursday, May 11, 2023
63. **Verify application uses transport layer protection if transmitting sensitive information, such as authenticated requests.** – AttackForge Admin on Thursday, May 11, 2023
64. **Verify application does not utilise self-signed certificate.** – AttackForge Admin on Thursday, May 11, 2023
65. **Verify that sensitive data is rapidly sanitized from memory as soon as itis no longer needed and handled in accordance to functions and techniques supported by the framework/library/operating system.** – AttackForge Admin on Thursday, May 11, 2023
66. **Verify that data stored in client side storage - such as HTML5 local storage, session storage, IndexedDB, regular cookies or Flash cookies - does not contain sensitive data or PII.** – AttackForge Admin on Thursday, May 11, 2023
67. **Verify that files obtained from untrusted sources are stored outside the webroot, with limited permissions, preferably with strong validation.**  – AttackForge Admin on Thursday, May 11, 2023
68. **Verify that all random numbers, random file titles, random GUIDs, and random strings are generated using the cryptographic module’s approved random number generator when these random values are intended to be not guessable by an attacker.** – AttackForge Admin on Thursday, May 11, 2023
69. **Verify that all API responses contain X-Content-Type-Options: nosniff and Content-Disposition: attachment; filetitle = 'api.json' (or other appropriate filetitle for the content type).** – AttackForge Admin on Thursday, May 11, 2023
70. **Verify that the application sets appropriate anti-caching headers as per the risk of the application, such as the following: Expires: Tue, 03 Jul 2001 06:00:00 GMT; Last-Modified: {now}; GMT Cache-Control: no-store, no-cache, must-revalidate, max-age=0; Cache-Control: post-check=0, pre-check=0; Pragma: no-cache.** – AttackForge Admin on Thursday, May 11, 2023
71. **Verify all pages and resources by default require authentication except those specifically intended to be public (Principle of complete mediation).** – AttackForge Admin on Saturday, April 15, 2023

A security principle that ensures that authority is not circumvented in subsequent requests of an object by a subject, by checking for authorization (rights and privileges) upon every request for the object.

In other words, the access requests by a subject for an object are completely mediated every time.

*“All accesses to objects must be checked to ensure that they are allowed.”*

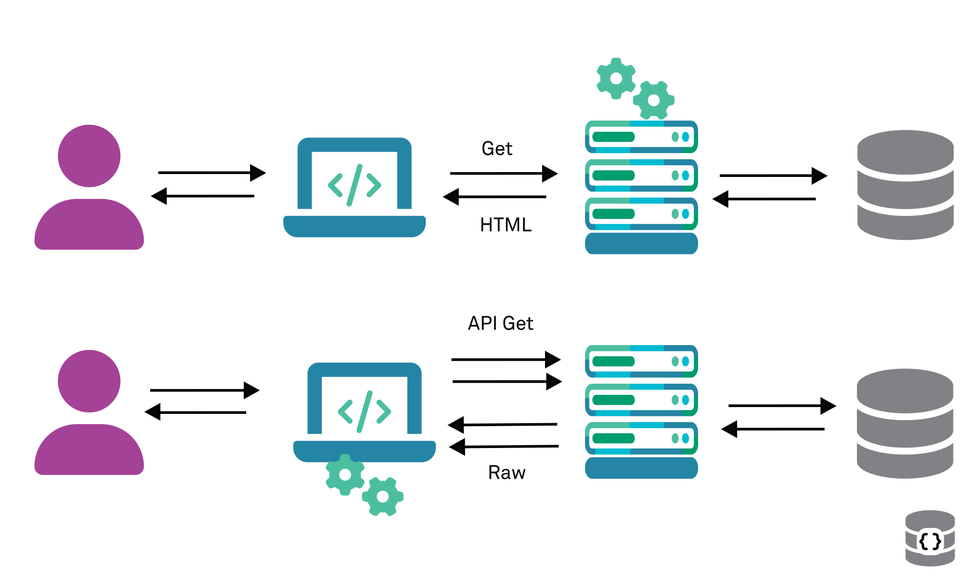
Performance v/s Security issue:

* Results of access checks are often cached
* What if permissions have changed since the last check?
* Mechanisms to invalidate or flush caches after a change are often missing.

09/30/2023 by AttackForge Admin

The following APIs were found to be accessible directly without any authentication token or credential:

* /api/users/{id}
* /api/users/{id}/profile
* /api/users/{id}/settings

  
Figure 7



unauthorized-access.png

1. **Verify that the message payload issigned to ensure reliable transport between client and service.**  – AttackForge Admin on Thursday, May 11, 2023
2. **Verify the application minimizes the number of parameters in a request, such as hidden fields, Ajax variables, cookies and header values.**  – AttackForge Admin on Thursday, May 11, 2023
3. **Verify all account identity authentication functions (such as update profile, forgot password, disabled / lost token, help desk or IVR) that might regain access to the account are at least as resistant to attack as the primary authentication mechanism.** – AttackForge Admin on Saturday, January 6, 2024
4. **Verify application does not utilise hardcoded credentials or passwords.** – AttackForge Admin on Thursday, May 11, 2023
5. **Verify that the application does not log sensitive data as defined underlocal privacy laws or regulations, organizational sensitive data as defined by a risk assessment, or sensitive authentication data that could assist an attacker, including user’s session identifiers, passwords, hashes, or API tokens.** – AttackForge Admin on Thursday, May 11, 2023
6. **Verify that the same access control rules implied by the presentation layer are enforced on the server side.**  – AttackForge Admin on Saturday, January 6, 2024
7. **Verify that only session ids generated by the application framework are recognized as active by the application.**  – AttackForge Admin on Saturday, January 6, 2024
8. **Verify that credentials are transported using a suitable encrypted link and that all pages/functions that require a user to enter credentials are done so using an encrypted link.** – AttackForge Admin on Saturday, April 15, 2023
9. **Verify that each log event includes necessary information that would allow for a detailed investigation of the timeline when an event happens.**  – AttackForge Admin on Thursday, May 11, 2023
10. **Verify that the TLS settings are in line with current leading practice, particularly as common configurations, ciphers, and algorithms become insecure.**  – AttackForge Admin on Thursday, May 11, 2023
11. **Verify that the application is not susceptible to LDAP Injection, or that security controls prevent LDAP Injection.**  – AttackForge Admin on Thursday, May 11, 2023
12. **Personally Identifiable Information should be stored encrypted at rest and ensure that communication goes via protected channels.**  – AttackForge Admin on Thursday, May 11, 2023
13. **Verify that access to administration and management functions within the application is limited to administrators.** – AttackForge Admin on Thursday, May 11, 2023
14. **Communications between components, such as between the application server and the database server should be authenticated using an account with the least necessary privileges.** – AttackForge Admin on Thursday, May 11, 2023
15. **Verify that client side validation is used as a second line of defense, in addition to server side validation.**  – AttackForge Admin on Thursday, May 11, 2023
16. **Verify that the system can be configured to disallow the use of a configurable number of previous passwords.**  – AttackForge Admin on Saturday, January 6, 2024
17. **Verify that where possible, keys and secrets are zeroed when destroyed.**  – AttackForge Admin on Thursday, May 11, 2023
18. **Verify that the application is not susceptible to Remote File Inclusion (RFI) or Local File Inclusion (LFI) when content is used that is a path to a file.**  – AttackForge Admin on Thursday, May 11, 2023
19. **Verify that sessions are invalidated when the user logs out.**  – AttackForge Admin on Saturday, April 15, 2023
20. **Verify that an active session list is displayed in the account profile orsimilar of each user. The user should be able to terminate any active session.** – AttackForge Admin on Saturday, January 6, 2024
21. **Verify application deployments are adequately sandboxed, containerized orisolated to delay and deter attackers from attacking other applications.**  – AttackForge Admin on Thursday, May 11, 2023
22. **Verify that error handling logic in security controls denies access by defau** – AttackForge Admin on Thursday, May 11, 2023
23. **Verify that there is an explicit policy for how cryptographic keys are managed (e.g., generated, distributed,revoked, and expired). Verify thatthis key lifecycle is properly enforced.** – AttackForge Admin on Thursday, May 11, 2023
24. **Verify that alternative and less secure access paths do not exist.**  – AttackForge Admin on Thursday, May 11, 2023
25. **Verify that the X-XSS-Protection: 1; mode=block header is in place.**  – AttackForge Admin on Thursday, May 11, 2023
26. **Verify that the application is protected from Cross-Site Request Forgery (CSRF).** – AttackForge Admin on Thursday, May 11, 2023
27. **All components should be up to date with proper security configuration(s) and version(s). This should include removal of unneeded configurations and folderssuch as sample applications, platform documentation, and default or example users.** – AttackForge Admin on Thursday, May 11, 2023
28. **Verify that HTTP Strict Transport Security headers are included on all requests and for all subdomains, such as Strict-Transport-Security: max-age=15724800; includeSubdomains** – AttackForge Admin on Thursday, May 11, 2023
29. **Verify that SOAP based web services are compliant with Web Services-Interoperability (WS-I) Basic Profile at minimum.** – AttackForge Admin on Thursday, May 11, 2023
30. **Verify the application has business limits and correctly enforces on a per user basis, with configurable alerting and automated reactions to automated or unusual attack.** – AttackForge Admin on Thursday, May 11, 2023
31. **Verify that all connections to external systems that involve sensitive information or functions are authenticated.**  – AttackForge Admin on Thursday, May 11, 2023
32. **Verify that URL redirects and forwards only allow whitelisted destinations, or show a warning when redirecting to potentially untrusted content.**  – AttackForge Admin on Thursday, May 11, 2023
33. **Verify that sessions timeout/expire after a specified period of inactivity** – AttackForge Admin on Saturday, January 6, 2024
34. **Verify that untrusted data is not used within inclusion, class loader, or reflection capabilities to prevent remote/local file inclusion vulnerabilities.**  – AttackForge Admin on Thursday, May 11, 2023
35. **Verify the application has additional authorization (such as step up or adaptive authentication) for lower value systems, and / orsegregation of dutiesfor high value applications to enforce anti-fraud controls as per the risk of application and past fraud.** – AttackForge Admin on Saturday, January 6, 2024
36. **Verify that if an application allows users to authenticate, they use a proven secure authentication mechanism** – AttackForge Admin on Saturday, January 6, 2024
37. **Verify that the same encoding style is used between the client and the server.**  – AttackForge Admin on Thursday, May 11, 2023
38. **Verify that the application accepts only a defined set of required HTTP request methods, such as GET and POST are accepted, and unused methods(e.g. TRACE, PUT, and DELETE) are explicitly blocked.** – AttackForge Admin on Thursday, May 11, 2023
39. **Verify application does not utilise predictable location for uploaded files.** – AttackForge Admin on Thursday, May 11, 2023
40. **Verify that access to sensitive records is protected, such that only authorised objects or data is accessible to each user (for example, protect against users tampering with a parameter to see or alter another user's account).** – AttackForge Admin on Saturday, January 6, 2024
41. **Verify that accountlockoutis divided into soft and hard lock status, and these are not mutually exclusive. If an account is temporarily soft locked out due to a brute force attack, this should not reset the hard lock status.** – AttackForge Admin on Saturday, January 6, 2024
42. **Verify that all cryptographic modules fail securely, and errors are handled in a way that does not enable oracle padding.**  – AttackForge Admin on Thursday, May 11, 2023
43. **Verify that untrusted file data submitted to the application is not used directly with file I/O commands, particularly to protect against path traversal, local file include, file mime type, and OS command injection vulnerabilities** – AttackForge Admin on Thursday, May 11, 2023
44. **Verify that the application does not output errormessages orstack traces containing sensitive data that could assist an attacker, including session id,software/framework versions and personal information.** – AttackForge Admin on Thursday, May 11, 2023
45. **Verify that structured data is strongly typed and validated against a defined schema including allowed characters, length and pattern (e.g. credit card numbers or telephone, or validating that two related fields are reasonable, such as validating suburbs and zip or post codes match).** – AttackForge Admin on Thursday, May 11, 2023
46. **Verify that all SQL queries, HQL, OSQL, NOSQL and stored procedures, calling of stored procedures are protected by the use of prepared statements or query parameterization, and thus not susceptible to SQL injection.** – AttackForge Admin on Thursday, May 11, 2023
47. **Verify that all authentication credentials for accessing services external to the application are encrypted and stored in a protected location.**  – AttackForge Admin on Saturday, January 6, 2024
48. **Verify the user is prompted with the option to terminate all other active sessions after a successful change password process.** – AttackForge Admin on Saturday, January 6, 2024
49. **Verify that data transferred from one DOM context to another, uses safe JavaScript methods, such as using .innerText and .val.**  – AttackForge Admin on Thursday, May 11, 2023
50. **Verify application enforces password security policy and/or requirements.** – AttackForge Admin on Thursday, May 11, 2023
51. **Verify that forgotten password and other recovery paths use a soft token, mobile push, or an offline recovery mechanism.**  – AttackForge Admin on Saturday, January 6, 2024
52. **Verify that untrusted data is not used within cross-domain resource sharing (CORS) to protect against arbitrary remote content.** – AttackForge Admin on Thursday, May 11, 2023
53. **Verify that all access control decisions can be logged and all failed decisions are logged.**  – AttackForge Admin on Saturday, January 6, 2024
54. **Verify that the HTTP headers or any part of the HTTP response do not expose detailed version information of system components.**  – AttackForge Admin on Thursday, May 11, 2023
55. **Verify application does not utilise third-party scripts from different origins.** – AttackForge Admin on Thursday, May 11, 2023
56. **Verify that the runtime environment is not susceptible to buffer overflows, or that security controls prevent buffer overflows.**  – AttackForge Admin on Saturday, January 6, 2024
57. **Verify that all keys and passwords are replaceable, and are generated orreplaced at installation time.** – AttackForge Admin on Thursday, May 11, 2023
58. **Verify all authentication controls are enforced on the server side.** – AttackForge Admin on Saturday, April 15, 2023
59. **Verify that session ids stored in cookies have their path set to an appropriately restrictive value for the application, and authentication session tokens additionally set the 'HttpOnly' and 'Secure' attributes** – AttackForge Admin on Saturday, January 6, 2024
60. **Verify that there is no custom session manager, orthatthe customsessionmanager is resistant against all common session management attacks.**  – AttackForge Admin on Saturday, January 6, 2024
61. **Verify that the application limits the number of active concurrent sessions.**  – AttackForge Admin on Saturday, January 6, 2024
62. **Verify that the application orframework usesstrong randomanti-CSRF tokens or has another transaction protection mechanism.**  – AttackForge Admin on Saturday, January 6, 2024
63. **Verify that request throttling is in place to prevent automated attacks against common authentication attacks such as brute force attacks or denial of service attacks.** – AttackForge Admin on Saturday, January 6, 2024
64. **Verify that the web or application serveris configured by defaultto deny access to remote resources or systems outside the web or application server.**  – AttackForge Admin on Thursday, May 11, 2023
65. **For auto-escaping template technology, if UI escaping is disabled, ensure thatHTML sanitization is enabled instead.**  – AttackForge Admin on Thursday, May 11, 2023
66. **Verify application server does not expose clear-text protocols for communication with server such as FTP, Telnet & SNMPv1 & 2.** – AttackForge Admin on Thursday, May 11, 2023
67. **Ensure forward secrecy ciphers are in use to mitigate passive attackers recording traffic.**  – AttackForge Admin on Thursday, May 11, 2023
68. **Verify the application explicitly checks the incoming Content-Type to be the expected one, such as application/xml or application/json.** – AttackForge Admin on Thursday, May 11, 2023
69. **Verify that the in-app purchase system cannot be cheated or bypassed.** – AttackForge Admin on Thursday, May 11, 2023

|  |
| --- |
| IN PROGRESS |
| NOT TESTED |
| NOT APPLICABLE |

1. **Verify that information enumeration is not possible via login, password reset, or forgot account functionality.**  – Bilbo Baggins on Saturday, April 15, 2023
2. **Verify that the changing password functionality includes the old password, the new password, and a password confirmation.**  – AttackForge Admin on Sunday, July 7, 2024



screenshot.png

# **UNIQUE VULNERABILITY DETAILS**

1. **Critical – Unrestricted Upload of File with Dangerous Type**

|  |  |
| --- | --- |
| **Unique Affected Assets** | **Unique Steps to Reproduce (POC)** |
| globexcorp.com.au | **Affected Assets**   * globexcorp.com.au   **POC**  How to reproduce   1. Do this... 2. Do that...   <some script>...<do something>...</some script>  image Figure 4 |

1. **High – Inconsistent Access Control**

|  |  |
| --- | --- |
| **Unique Affected Assets** | **Unique Steps to Reproduce (POC)** |
| globexcorp.com.au  https://globexcorp.com.au/endpoint5 | **Affected Assets**   * globexcorp.com.au   **Affected Endpoints**   * https://globexcorp.com.au/endpoint5   **POC**  - Login in to the application as Standard user  - Once in the dashboard, take note and copy the value of the 'X' visible from within the URL  - Note that the administrative tab does not appear when clicking on the icon on the top right-hand side as an standard user role is used  - Replace the value of 'X' in the following URL with the copied one and browse to it:globexcorp.com.au/crm/test/page?X=6365973057477497141868253726 |

1. **High – Relative Path Traversal**

|  |  |
| --- | --- |
| **Unique Affected Assets** | **Unique Steps to Reproduce (POC)** |
| globexcorp.com.au  https://globexcorp.com.au/endpoint2 | **Affected Assets**   * globexcorp.com.au   **Affected Endpoints**   * https://globexcorp.com.au/endpoint2   **POC**  Preconditions  Authenticate to the portal as administrator user and browse to the following URL:  https://globexcorp.com.au/test/cmsedit.jsp?file=../../../../../../../../etc/hostname  Note that the resulting page will contain the hostname of the underlying system.  Affected   * URL: https://globexcorp.com.au/test/cmsedit.jsp?file=../../../../../../../../etc/hostname * Parameter: file |

1. **High – Reflected Cross Site Scripting**

|  |  |
| --- | --- |
| **Unique Affected Assets** | **Unique Steps to Reproduce (POC)** |
| globexcorp.com.au  https://globexcorp.com.au/main | **Affected Assets**   * globexcorp.com.au   **Affected Endpoints**   * https://globexcorp.com.au/main   **POC**  1. do this..  2. do this.. |

1. **High – Blind SQL Injection**

|  |  |
| --- | --- |
| **Unique Affected Assets** | **Unique Steps to Reproduce (POC)** |
| globexcorp.com.au  https://globexcorp.com.au/endpoint1 | **Affected Assets**   * globexcorp.com.au   **Affected Endpoints**   * https://globexcorp.com.au/endpoint1   **POC**  It is possible to verify the existence of this issue by visiting the following URL and observing the that the page returned the application for user 597098 successfully:  - https://globexcorp.com.au/crm/test/ReportBase.jsp?UserId=597098%20and%203=3  Note the result won't include the report anymore. This is because the logic operand injected in the SQL query is AND 3=1 which returns false. In turn the database will return false to the application and did not return any data. This is a simple example that demonstrates the logic of SQL injection and operands. |

1. **High – Persistent Cross Site Scripting**

|  |  |
| --- | --- |
| **Unique Affected Assets** | **Unique Steps to Reproduce (POC)** |
| globexcorp.com.au  https://globexcorp.com.au/endpoint5 | **Affected Assets**   * globexcorp.com.au   **Affected Endpoints**   * https://globexcorp.com.au/endpoint5   **POC**  To reproduce this issue, open an existing or create a new application from the 'New Application' menu. Then access the Notes section from the submenu on the right hand side of the page. Fill the subject field with 'test' and the message field with 'alert('XSS')' then save. The page will automatically reload resulting in a a pop up message with 'XSS' as text. This type of JavaScript code is benign, however an attacker would use malicious code that can attack other users of the application, for example to steal their session cookie/token. |

1. **High – Unsupported Web Server Detection**

|  |  |
| --- | --- |
| **Unique Affected Assets** | **Unique Steps to Reproduce (POC)** |
| form.acmegroup.co  classic.acmegroup.co  54.215.221.230:443  13.56.222.64:443 | **Affected Assets**   * form.acmegroup.co * classic.acmegroup.co   **Affected Endpoints**   * 54.215.221.230:443 * 13.56.222.64:443   **POC**  Product : Apache 2.2.x  Server response header : Apache/2.2.34 (Amazon)  Supported versions : Apache HTTP Server 2.4.x  Additional information : http://archive.apache.org/dist/httpd/Announcement2.2.html |

1. **High – PHP Unsupported Version Detection**

|  |  |
| --- | --- |
| **Unique Affected Assets** | **Unique Steps to Reproduce (POC)** |
| classic.acmegroup.co  13.56.222.64:443 | **Affected Assets**   * classic.acmegroup.co   **Affected Endpoints**   * 13.56.222.64:443   **POC**  Source : X-Powered-By: PHP/5.3.29  Installed version : 5.3.29  End of support date : 2014/08/14  Announcement : http://php.net/archive/2014.php#id2014-08-14-1  Supported versions : 7.1.x / 7.2.x / 7.3.x |

1. **High – SSL Version 2 and 3 Protocol Detection**

|  |  |
| --- | --- |
| **Unique Affected Assets** | **Unique Steps to Reproduce (POC)** |
| classic.acmegroup.co  52.52.53.216:443 | **Affected Assets**   * classic.acmegroup.co   **Affected Endpoints**   * 52.52.53.216:443   **POC**  - SSLv3 is enabled and the server supports at least one cipher.  Explanation: TLS 1.0 and SSL 3.0 cipher suites may be used with SSLv3    Medium Strength Ciphers (> 64-bit and < 112-bit key, or 3DES)   EDH-RSA-DES-CBC3-SHA Kx=DH Au=RSA Enc=3DES-CBC(168) Mac=SHA1   ECDHE-RSA-DES-CBC3-SHA Kx=ECDH Au=RSA Enc=3DES-CBC(168) Mac=SHA1   DES-CBC3-SHA Kx=RSA Au=RSA Enc=3DES-CBC(168) Mac=SHA1    High Strength Ciphers (>= 112-bit key)   DHE-RSA-AES128-SHA Kx=DH Au=RSA Enc=AES-CBC(128) Mac=SHA1   DHE-RSA-AES256-SHA Kx=DH Au=RSA Enc=AES-CBC(256) Mac=SHA1   DHE-RSA-CAMELLIA128-SHA Kx=DH Au=RSA Enc=Camellia-CBC(128) Mac=SHA1   DHE-RSA-CAMELLIA256-SHA Kx=DH Au=RSA Enc=Camellia-CBC(256) Mac=SHA1   DHE-RSA-SEED-SHA Kx=DH Au=RSA Enc=SEED-CBC(128) Mac=SHA1   ECDHE-RSA-AES128-SHA Kx=ECDH Au=RSA Enc=AES-CBC(128) Mac=SHA1   ECDHE-RSA-AES256-SHA Kx=ECDH Au=RSA Enc=AES-CBC(256) Mac=SHA1   AES128-SHA Kx=RSA Au=RSA Enc=AES-CBC(128) Mac=SHA1   AES256-SHA Kx=RSA Au=RSA Enc=AES-CBC(256) Mac=SHA1   CAMELLIA128-SHA Kx=RSA Au=RSA Enc=Camellia-CBC(128) Mac=SHA1   CAMELLIA256-SHA Kx=RSA Au=RSA Enc=Camellia-CBC(256) Mac=SHA1   SEED-SHA Kx=RSA Au=RSA Enc=SEED-CBC(128) Mac=SHA1   DHE-RSA-AES128-SHA256 Kx=DH Au=RSA Enc=AES-CBC(128) Mac=SHA256   DHE-RSA-AES256-SHA256 Kx=DH Au=RSA Enc=AES-CBC(256) Mac=SHA256   ECDHE-RSA-AES128-SHA256 Kx=ECDH Au=RSA Enc=AES-CBC(128) Mac=SHA256   ECDHE-RSA-AES256-SHA384 Kx=ECDH Au=RSA Enc=AES-CBC(256) Mac=SHA384   RSA-AES128-SHA256 Kx=RSA Au=RSA Enc=AES-CBC(128) Mac=SHA256   RSA-AES256-SHA256 Kx=RSA Au=RSA Enc=AES-CBC(256) Mac=SHA256   The fields above are :   {OpenSSL ciphername}  Kx={key exchange}  Au={authentication}  Enc={symmetric encryption method}  Mac={message authentication code}  {export flag} |

1. **Medium – Weak Password Policy**

|  |  |
| --- | --- |
| **Unique Affected Assets** | **Unique Steps to Reproduce (POC)** |
| globexcorp.com.au  https://globexcorp.com.au/login | **Affected Assets**   * globexcorp.com.au   **Affected Endpoints**   * https://globexcorp.com.au/login   **POC**  Through the admin panel, change a user's password to a known weak password such as 'password' or '1234'. Take note that the password change is successful. The login form enforces the use of passwords that are longer than 8 characters, however this restriction only applies to the input box on the login form and not the authentication endpoint. Therefore it is possible to submit a valid password of less than 8 characters to the authentication endpoint and be successfully logged in, making brute force attacks possible against weak passwords. |

1. **Medium – Session Fixation**

|  |  |
| --- | --- |
| **Unique Affected Assets** | **Unique Steps to Reproduce (POC)** |
| globexcorp.com.au  https://globexcorp.com.au/\* | **Affected Assets**   * globexcorp.com.au   **Affected Endpoints**   * https://globexcorp.com.au/\*   **POC**  1. Using an intercepting proxy or browser development tools, login as user 'X' and generate a case report.  2. Take note of the value for the 'X' cookie.  3. Log out and clear all browser cache.  4. Repeat step 1.  5. Take note the value for the 'X' cookie is same as previous i.e. static. |

1. **Medium – SSL Weak Ciphers**

|  |  |
| --- | --- |
| **Unique Affected Assets** | **Unique Steps to Reproduce (POC)** |
| globexcorp.com.au  http://10.0.0.1:443/ | **Affected Assets**   * globexcorp.com.au   **Affected Endpoints**   * http://10.0.0.1:443/   **POC**  Use a tool such ass sslscan to enumerate the ciphers supported by the application server. |

1. **Medium – Reflected Cross Site Scripting**

|  |  |
| --- | --- |
| **Unique Affected Assets** | **Unique Steps to Reproduce (POC)** |
| globexcorp.com.au  classic.acmegroup.co  https://globexcorp.com.au/profile  sdf | **Affected Assets**   * globexcorp.com.au   **Affected Endpoints**   * https://globexcorp.com.au/profile   **POC**  Using a web proxy or a command line tool (such as curl or wget), submit the request attached to this finding. Notice the response contains a 'Content-Type: text/html;', and includes the HTML payload passed in the URL.  **Affected Assets**   * classic.acmegroup.co   **POC**   1. do this. 2. do that.   <script>alert(1)</script>    **Affected Assets**   * classic.acmegroup.co   **Affected Endpoints**   * sdf   **POC**  do this... |

1. **Medium – Cross Site Request Forgery (CSRF)**

|  |  |
| --- | --- |
| **Unique Affected Assets** | **Unique Steps to Reproduce (POC)** |
| globexcorp.com.au  https://globexcorp.com.au/profile | **Affected Assets**   * globexcorp.com.au   **Affected Endpoints**   * https://globexcorp.com.au/profile   **POC**  1. Open in a web browser the csrf-test.html proof-of-concept file uploaded. The form will be submitted without the users consent.  2. Open the cases tab in the app: /crm/cases  3. You will find the new cases on top of the list. |

1. **Medium – Functionality Misuse**

|  |  |
| --- | --- |
| **Unique Affected Assets** | **Unique Steps to Reproduce (POC)** |
| globexcorp.com.au  https://globexcorp.com.au/endpoint2 | **Affected Assets**   * globexcorp.com.au   **Affected Endpoints**   * https://globexcorp.com.au/endpoint2   **POC**  It is possible to create a case on behalf of another user, or a non-existing-user, by modifying the 'X' parameter to another email address during request to POST /crm/test?encoding=UTF endpoint. |

1. **Medium – No Account Lockout May Facilitate Brute Force Password Attack**

|  |  |
| --- | --- |
| **Unique Affected Assets** | **Unique Steps to Reproduce (POC)** |
| globexcorp.com.au  https://globexcorp.com.au/login | **Affected Assets**   * globexcorp.com.au   **Affected Endpoints**   * https://globexcorp.com.au/login   **POC**  Attempt to authenticate with the wrong password more than ten (10) times in the application, then login with the correct password. Note that the application did not lock the user account although numerous attempts were performed. . |

1. **Medium – Open Redirection**

|  |  |
| --- | --- |
| **Unique Affected Assets** | **Unique Steps to Reproduce (POC)** |
| globexcorp.com.au  https://globexcorp.com.au/login | **Affected Assets**   * globexcorp.com.au   **Affected Endpoints**   * https://globexcorp.com.au/login   **POC**  1. Using an intercepting proxy, modify the 'retURL' parameter for the POST /crm/test?encoding=UTF-8 endpoint to another URL, for instance https://globexcorp.com.au.   2. Take note after form is processed, the user is redirected to attacker's website i.e. https://globexcorp.com.au. |

1. **Medium – SSL Certificate Expiry**

|  |  |
| --- | --- |
| **Unique Affected Assets** | **Unique Steps to Reproduce (POC)** |
| classic.acmegroup.co  13.56.222.64:443 | **Affected Assets**   * classic.acmegroup.co   **Affected Endpoints**   * 13.56.222.64:443   **POC**  The SSL certificate has already expired :   Subject : C=XX, L=Default City, O=Default Company Ltd  Issuer : C=XX, L=Default City, O=Default Company Ltd  Not valid before : Sep 27 11:01:37 2017 GMT  Not valid after : Sep 27 11:01:37 2018 GMT |

1. **Medium – SSL Weak Cipher Suites Supported**

|  |  |
| --- | --- |
| **Unique Affected Assets** | **Unique Steps to Reproduce (POC)** |
| classic.acmegroup.co  13.56.222.64:443 | **Affected Assets**   * classic.acmegroup.co   **Affected Endpoints**   * 13.56.222.64:443   **POC**  Here is the list of weak SSL ciphers supported by the remote server :   Low Strength Ciphers (<= 64-bit key)   EDH-RSA-DES-CBC-SHA Kx=DH Au=RSA Enc=DES-CBC(56) Mac=SHA1   DES-CBC-SHA Kx=RSA Au=RSA Enc=DES-CBC(56) Mac=SHA1   The fields above are :   {OpenSSL ciphername}  Kx={key exchange}  Au={authentication}  Enc={symmetric encryption method}  Mac={message authentication code}  {export flag} |

1. **Medium – SSL Certificate Signed Using Weak Hashing Algorithm**

|  |  |
| --- | --- |
| **Unique Affected Assets** | **Unique Steps to Reproduce (POC)** |
| classic.acmegroup.co  13.56.222.64:443 | **Affected Assets**   * classic.acmegroup.co   **Affected Endpoints**   * 13.56.222.64:443   **POC**  The following certificates were part of the certificate chain sent by the remote host, but contain hashes that are considered to be weak.  |-Subject : C=XX/L=Default City/O=Default Company Ltd |-Signature Algorithm : SHA-1 With RSA Encryption |-Valid From : Sep 27 11:01:37 2017 GMT |-Valid To : Sep 27 11:01:37 2018 GMT |

1. **Medium – SSL Medium Strength Cipher Suites Supported (SWEET32)**

|  |  |
| --- | --- |
| **Unique Affected Assets** | **Unique Steps to Reproduce (POC)** |
| classic.acmegroup.co  13.56.222.64:443  52.52.53.216:443 | **Affected Assets**   * classic.acmegroup.co   **Affected Endpoints**   * 13.56.222.64:443 * 52.52.53.216:443   **POC**  Medium Strength Ciphers (> 64-bit and < 112-bit key, or 3DES)   EDH-RSA-DES-CBC3-SHA Kx=DH Au=RSA Enc=3DES-CBC(168) Mac=SHA1   ECDHE-RSA-DES-CBC3-SHA Kx=ECDH Au=RSA Enc=3DES-CBC(168) Mac=SHA1   DES-CBC3-SHA Kx=RSA Au=RSA Enc=3DES-CBC(168) Mac=SHA1   The fields above are :   {OpenSSL ciphername}  Kx={key exchange}  Au={authentication}  Enc={symmetric encryption method}  Mac={message authentication code}  {export flag} |

1. **Medium – SSL Certificate Cannot Be Trusted**

|  |  |
| --- | --- |
| **Unique Affected Assets** | **Unique Steps to Reproduce (POC)** |
| classic.acmegroup.co  13.56.222.64:443  54.153.59.11:443 | **Affected Assets**   * classic.acmegroup.co   **Affected Endpoints**   * 13.56.222.64:443   **POC**  The following certificate was part of the certificate chain sent by the remote host, but it has expired :  |-Subject : C=XX/L=Default City/O=Default Company Ltd |-Not After : Sep 27 11:01:37 2018 GMT  The following certificate was at the top of the certificate chain sent by the remote host, but it is signed by an unknown certificate authority :  |-Subject : C=XX/L=Default City/O=Default Company Ltd |-Issuer : C=XX/L=Default City/O=Default Company Ltd  **Affected Assets**   * classic.acmegroup.co   **Affected Endpoints**   * 54.153.59.11:443   **POC**  The following certificate was at the top of the certificate chain sent by the remote host, but it is signed by an unknown certificate authority :  |-Subject : C=au/L=admin/O=acme/CN=acme WebAdmin CA/E=dan@acmegroup.co |-Issuer : C=au/L=admin/O=acme/CN=acme WebAdmin CA/E=dan@acmegroup.co |

1. **Medium – SSL Self-Signed Certificate**

|  |  |
| --- | --- |
| **Unique Affected Assets** | **Unique Steps to Reproduce (POC)** |
| classic.acmegroup.co  13.56.222.64:443  54.153.59.11:443 | **Affected Assets**   * classic.acmegroup.co   **Affected Endpoints**   * 13.56.222.64:443   **POC**  The following certificate was found at the top of the certificate chain sent by the remote host, but is self-signed and was not found in the list of known certificate authorities :  |-Subject : C=XX/L=Default City/O=Default Company Ltd  **Affected Assets**   * classic.acmegroup.co   **Affected Endpoints**   * 54.153.59.11:443   **POC**  The following certificate was found at the top of the certificate chain sent by the remote host, but is self-signed and was not found in the list of known certificate authorities :  |-Subject : C=au/L=admin/O=acme/CN=acme WebAdmin CA/E=dan@acmegroup.co |

1. **Low – Multiple Simultaneous User Sessions**

|  |  |
| --- | --- |
| **Unique Affected Assets** | **Unique Steps to Reproduce (POC)** |
| globexcorp.com.au  https://globexcorp.com.au/\* | **Affected Assets**   * globexcorp.com.au   **Affected Endpoints**   * https://globexcorp.com.au/\*   **POC**  Access the application in multiple browsers and take note that simultaneous sessions are permitted. Also take note that no login information for the previous session is displayed to the user on the login landing page. |

1. **Low – HTML 5 Cross Origin Resource Sharing**

|  |  |
| --- | --- |
| **Unique Affected Assets** | **Unique Steps to Reproduce (POC)** |
| globexcorp.com.au  https://globexcorp.com.au/\* | **Affected Assets**   * globexcorp.com.au   **Affected Endpoints**   * https://globexcorp.com.au/\*   **POC**  Access the application with a browser and an intercepting proxy. Take note that the application allows cross origin resource sharing for all origins, specified by the asterisk on API calls. |

1. **Low – Missing X-XSS-Protection Header**

|  |  |
| --- | --- |
| **Unique Affected Assets** | **Unique Steps to Reproduce (POC)** |
| globexcorp.com.au  https://globexcorp.com.au/\* | **Affected Assets**   * globexcorp.com.au   **Affected Endpoints**   * https://globexcorp.com.au/\*   **POC**  Access the application with a browser and an intercepting proxy. Take note that the application does not set the X-XSS-Protection header in HTTP responses. |

1. **Low – Cookie Without HTTPOnly Flag Set**

|  |  |
| --- | --- |
| **Unique Affected Assets** | **Unique Steps to Reproduce (POC)** |
| globexcorp.com.au  https://globexcorp.com.au/\* | **Affected Assets**   * globexcorp.com.au   **Affected Endpoints**   * https://globexcorp.com.au/\*   **POC**  To reproduce this issue configure an intercepting proxy and monitor application traffic. Note the absence of HTTPOnly flag in set-cookie header. |

1. **Low – Strict Transport Security Policy Not Enforced**

|  |  |
| --- | --- |
| **Unique Affected Assets** | **Unique Steps to Reproduce (POC)** |
| globexcorp.com.au  https://globexcorp.com.au/\* | **Affected Assets**   * globexcorp.com.au   **Affected Endpoints**   * https://globexcorp.com.au/\*   **POC**  This vulnerability can be observed from the HTTP response of the application server, note HSTS header is missing. This security feature can be enabled by editing the web.config file. |

1. **Low – Outdated Version of JQuery**

|  |  |
| --- | --- |
| **Unique Affected Assets** | **Unique Steps to Reproduce (POC)** |
| globexcorp.com.au  https://globexcorp.com.au/scripts.js | **Affected Assets**   * globexcorp.com.au   **Affected Endpoints**   * https://globexcorp.com.au/scripts.js   **POC**  Access any of the listed files below and note the jQuery version mentioned on top of the file. - /crm/test/file1.html - /crm/test/file2.html - /crm/test/file3.html |

1. **Low – Cookie With Secure Flag Missing**

|  |  |
| --- | --- |
| **Unique Affected Assets** | **Unique Steps to Reproduce (POC)** |
| globexcorp.com.au  https://globexcorp.com.au/\* | **Affected Assets**   * globexcorp.com.au   **Affected Endpoints**   * https://globexcorp.com.au/\*   **POC**  To reproduce this issue configure an intercepting proxy and monitor application traffic. Note the absence of secure flag in set-cookie header. |

1. **Low – User Enumeration**

|  |  |
| --- | --- |
| **Unique Affected Assets** | **Unique Steps to Reproduce (POC)** |
| globexcorp.com.au  https://globexcorp.com.au/login | **Affected Assets**   * globexcorp.com.au   **Affected Endpoints**   * https://globexcorp.com.au/login   **POC**  Try to sign up for a new user. Supply the sign up form with an email that belongs to an existing account, and notice the application displays a different error message saying an account already exists with the email. |

1. **Low – Server Discloses Supporting Technology**

|  |  |
| --- | --- |
| **Unique Affected Assets** | **Unique Steps to Reproduce (POC)** |
| globexcorp.com.au  https://globexcorp.com.au/\* | **Affected Assets**   * globexcorp.com.au   **Affected Endpoints**   * https://globexcorp.com.au/\*   **POC**  Access the application with a browser and an intercepting proxy. Take note that the application discloses the software and version of nginx in use in HTTP responses. |

1. **Low – Insufficient Framing Protection**

|  |  |
| --- | --- |
| **Unique Affected Assets** | **Unique Steps to Reproduce (POC)** |
| globexcorp.com.au  https://globexcorp.com.au/\* | **Affected Assets**   * globexcorp.com.au   **Affected Endpoints**   * https://globexcorp.com.au/\*   **POC**  Access the application with an intercepting proxy and a browser. Take note that the application does not set X-Frame-Options headers in HTTP responses. |

1. **Low – Insufficient Session Expiration**

|  |  |
| --- | --- |
| **Unique Affected Assets** | **Unique Steps to Reproduce (POC)** |
| globexcorp.com.au  https://globexcorp.com.au/\* | **Affected Assets**   * globexcorp.com.au   **Affected Endpoints**   * https://globexcorp.com.au/\*   **POC**  Using an intercepting proxy, replay API calls to the application after a user has logged out. Take note that the session token remains active. |

1. **Low – SSL RC4 Cipher Suites Supported (Bar Mitzvah)**

|  |  |
| --- | --- |
| **Unique Affected Assets** | **Unique Steps to Reproduce (POC)** |
| form.acmegroup.co  13.56.222.64:443 | **Affected Assets**   * form.acmegroup.co   **Affected Endpoints**   * 13.56.222.64:443   **POC**  List of RC4 cipher suites supported by the remote server :   High Strength Ciphers (>= 112-bit key)   ECDHE-RSA-RC4-SHA Kx=ECDH Au=RSA Enc=RC4(128) Mac=SHA1   RC4-MD5 Kx=RSA Au=RSA Enc=RC4(128) Mac=MD5   RC4-SHA Kx=RSA Au=RSA Enc=RC4(128) Mac=SHA1   The fields above are :   {OpenSSL ciphername}  Kx={key exchange}  Au={authentication}  Enc={symmetric encryption method}  Mac={message authentication code}  {export flag} |

1. **Info – Inducing Account Lockout**

|  |  |
| --- | --- |
| **Unique Affected Assets** | **Unique Steps to Reproduce (POC)** |
| globexcorp.com.au  https://globexcorp.com.au/login | **Affected Assets**   * globexcorp.com.au   **Affected Endpoints**   * https://globexcorp.com.au/login   **POC**  1. do this..  2. do that.. |

1. **Info – TLS Version 1.1 Protocol Detection**

|  |  |
| --- | --- |
| **Unique Affected Assets** | **Unique Steps to Reproduce (POC)** |
| classic.acmegroup.co  52.8.11.65:443 | **Affected Assets**   * classic.acmegroup.co   **Affected Endpoints**   * 52.8.11.65:443   **POC**  TLSv1.1 is enabled and the server supports at least one cipher. |

1. **Info – nginx HTTP Server Detection**

|  |  |
| --- | --- |
| **Unique Affected Assets** | **Unique Steps to Reproduce (POC)** |
| classic.acmegroup.co  52.8.11.65:443 | **Affected Assets**   * classic.acmegroup.co   **Affected Endpoints**   * 52.8.11.65:443   **POC**  URL : https://classic.acmegroup.co/  Version : 1.10.2  source : Server: nginx/1.10.2 |

1. **Info – TLS Version 1.0 Protocol Detection**

|  |  |
| --- | --- |
| **Unique Affected Assets** | **Unique Steps to Reproduce (POC)** |
| classic.acmegroup.co  52.8.11.65:443 | **Affected Assets**   * classic.acmegroup.co   **Affected Endpoints**   * 52.8.11.65:443   **POC**  TLSv1 is enabled and the server supports at least one cipher. |