VALIDADORAPORTACIONESVOLUNTARIAS Reporte del escaneo

Nombre del proyecto	VALIDADORAPORTACIONESVOLUNTARIAS
Iniciar Escaneo	miércoles, 14 de diciembre de 2022 10:06:54
Conjunto de Consultas	Coppel Default Test
Tiempo de escaneo	00h:03m:41s
Líneas de código escaneadas	42786
Archivos escaneados	201
Hora de creación del reporte	miércoles, 14 de diciembre de 2022 10:34:09
Resultados en línea	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1155844&projectid=45512
Equipo	Proy_Remediaciones
Versión de Checkmarx	9.5.1.1001 HF3
Tipo de escaneo	Completo
Origen de la Fuente	LocalPath
Densidad	5/10000 (Vulnerabilidades/LOC)
Visibilidad	Público

Configuración de filtro

Severidad

Incluido: Altas, Medias, Bajas, Información

Excluido: Ninguna

Estado del resultado

Incluido: Para verificar, No explotable, Confirmado, Urgente, Propuesto no explotable

Excluido: Ninguna

Asignado a

Incluido: Todas

Categorías

Incluido:

Sin categoría	Todas
Custom	Todas
PCI DSS v3.2.1	Todas
OWASP Top 10 2013	Todas
FISMA 2014	Todas
NIST SP 800-53	Todas
OWASP Top 10 2017	Todas
OWASP Mobile Top 10 2016	Todas
OWASP Top 10 API	Todas
ASD STIG 4.10	Todas
OWASP Top 10 2010	Todas
OWASP Top 10 2021	Todas
MOIS(KISA) Secure	Todas

Coding 2021	
SANS top 25	Todas
CWE top 25	Todas
OWASP ASVS	Todas

Excluido:

Sin categoría	Ninguna
Custom	Ninguna
PCI DSS v3.2.1	Ninguna
OWASP Top 10 2013	Ninguna
FISMA 2014	Ninguna
NIST SP 800-53	Ninguna
OWASP Top 10 2017	Ninguna
OWASP Mobile Top 10 2016	Ninguna
OWASP Top 10 API	Ninguna
ASD STIG 4.10	Ninguna
OWASP Top 10 2010	Ninguna
OWASP Top 10 2021	Ninguna
MOIS(KISA) Secure Coding 2021	Ninguna
SANS top 25	Ninguna
CWE top 25	Ninguna
OWASP ASVS	Ninguna

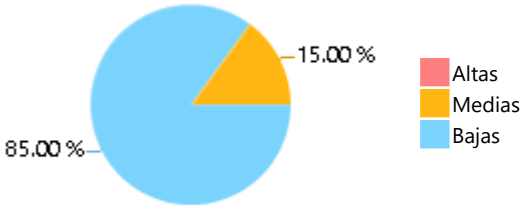
Límite de resultados

El límite de resultados por consulta se estableció en 50

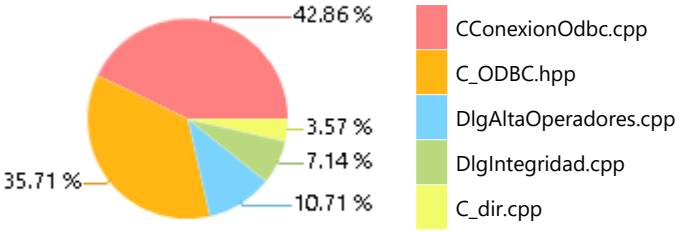
Consultas seleccionadas

Las consultas seleccionadas estan listadas en Resumen de los resultados

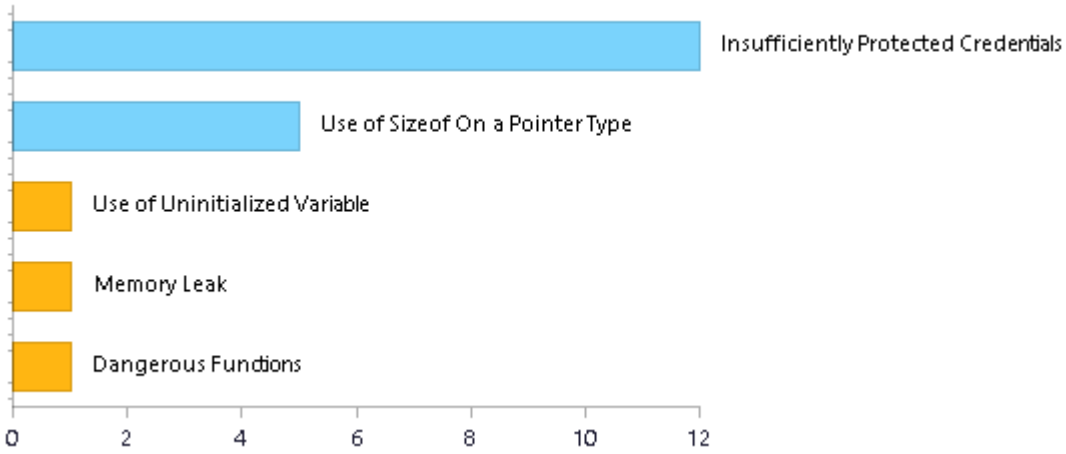
Resumen de los resultados



Archivos más vulnerables



Las 5 Vulnerabilidades Principales



Resumen de escaneo - OWASP Top 10 2017

Información adicional sobre vulnerabilidades y riesgos puede ser encontrada en: OWASP Top 10 2017

Category	Threat Agent	Exploitability	Weakness Prevalence	Weakness Detectability	Technical Impact	Business Impact	Issues Found	Best Fix Locations
A1-Injection*	App. Specific	EASY	COMMON	EASY	SEVERE	App. Specific	0	0
A2-Broken Authentication	App. Specific	EASY	COMMON	AVERAGE	SEVERE	App. Specific	0	0
A3-Sensitive Data Exposure*	App. Specific	AVERAGE	WIDESPREAD	AVERAGE	SEVERE	App. Specific	12	4
A4-XML External Entities (XXE)	App. Specific	AVERAGE	COMMON	EASY	SEVERE	App. Specific	0	0
A5-Broken Access Control*	App. Specific	AVERAGE	COMMON	AVERAGE	SEVERE	App. Specific	0	0
A6-Security Misconfiguration	App. Specific	EASY	WIDESPREAD	EASY	MODERATE	App. Specific	0	0
A7-Cross-Site Scripting (XSS)*	App. Specific	EASY	WIDESPREAD	EASY	MODERATE	App. Specific	0	0
A8-Insecure Deserialization	App. Specific	DIFFICULT	COMMON	AVERAGE	SEVERE	App. Specific	0	0
A9-Using Components with Known Vulnerabilities*	App. Specific	AVERAGE	WIDESPREAD	AVERAGE	MODERATE	App. Specific	1	1
A10-Insufficient Logging & Monitoring	App. Specific	AVERAGE	WIDESPREAD	DIFFICULT	MODERATE	App. Specific	0	0

***** Los resultados del escaneo del proyecto no incluyen todas las consultas relevantes. El grupo de consultas y/o los filtros deben ser cambiados para incluir todas las consultas estandard

Resumen de escaneo - OWASP Top 10 2021

Category	Issues Found	Best Fix Locations
A1-Broken Access Control*	0	0
A2-Cryptographic Failures*	0	0
A3-Injection*	0	0
A4-Insecure Design*	14	6
A5-Security Misconfiguration*	0	0
A6-Vulnerable and Outdated Components*	0	0
A7-Identification and Authentication Failures*	0	0
A8-Software and Data Integrity Failures*	0	0
A9-Security Logging and Monitoring Failures*	0	0
A10-Server-Side Request Forgery	0	0

***** Los resultados del escaneo del proyecto no incluyen todas las consultas relevantes. El grupo de consultas y/o los filtros deben ser cambiados para incluir todas las consultas estandard

Resumen de escaneo - OWASP Top 10 2013

Información adicional sobre vulnerabilidades y riesgos puede ser encontrada en: OWASP Top 10 2013

Category	Threat Agent	Attack Vectors	Weakness Prevalence	Weakness Detectability	Technical Impact	Business Impact	Issues Found	Best Fix Locations
A1-Injection	EXTERNAL, INTERNAL, ADMIN USERS	EASY	COMMON	AVERAGE	SEVERE	ALL DATA	0	0
A2-Broken Authentication and Session Management	EXTERNAL, INTERNAL USERS	AVERAGE	WIDESPREAD	AVERAGE	SEVERE	AFFECTED DATA AND FUNCTIONS	0	0
A3-Cross-Site Scripting (XSS)*	EXTERNAL, INTERNAL, ADMIN USERS	AVERAGE	VERY WIDESPREAD	EASY	MODERATE	AFFECTED DATA AND SYSTEM	0	0
A4-Insecure Direct Object References*	SYSTEM USERS	EASY	COMMON	EASY	MODERATE	EXPOSED DATA	0	0
A5-Security Misconfiguration	EXTERNAL, INTERNAL, ADMIN USERS	EASY	COMMON	EASY	MODERATE	ALL DATA AND SYSTEM	0	0
A6-Sensitive Data Exposure*	EXTERNAL, INTERNAL, ADMIN USERS, USERS BROWSERS	DIFFICULT	UNCOMMON	AVERAGE	SEVERE	EXPOSED DATA	12	4
A7-Missing Function Level Access Control*	EXTERNAL, INTERNAL USERS	EASY	COMMON	AVERAGE	MODERATE	EXPOSED DATA AND FUNCTIONS	0	0
A8-Cross-Site Request Forgery (CSRF)*	USERS BROWSERS	AVERAGE	COMMON	EASY	MODERATE	AFFECTED DATA AND FUNCTIONS	0	0
A9-Using Components with Known Vulnerabilities*	EXTERNAL USERS, AUTOMATED TOOLS	AVERAGE	WIDESPREAD	DIFFICULT	MODERATE	AFFECTED DATA AND FUNCTIONS	1	1
A10-Unvalidated Redirects and Forwards	USERS BROWSERS	AVERAGE	WIDESPREAD	DIFFICULT	MODERATE	AFFECTED DATA AND FUNCTIONS	0	0

***** Los resultados del escaneo del proyecto no incluyen todas las consultas relevantes. El grupo de consultas y/o los filtros deben ser cambiados para incluir todas las consultas estandard

Resumen de escaneo - PCI DSS v3.2.1

Category	Issues Found	Best Fix Locations
PCI DSS (3.2.1) - 6.5.1 - Injection flaws - particularly SQL injection	0	0
PCI DSS (3.2.1) - 6.5.2 - Buffer overflows*	0	0
PCI DSS (3.2.1) - 6.5.3 - Insecure cryptographic storage	0	0
PCI DSS (3.2.1) - 6.5.4 - Insecure communications	0	0
PCI DSS (3.2.1) - 6.5.5 - Improper error handling*	0	0
PCI DSS (3.2.1) - 6.5.7 - Cross-site scripting (XSS)	0	0
PCI DSS (3.2.1) - 6.5.8 - Improper access control	0	0
PCI DSS (3.2.1) - 6.5.9 - Cross-site request forgery*	0	0
PCI DSS (3.2.1) - 6.5.10 - Broken authentication and session management*	0	0

***** Los resultados del escaneo del proyecto no incluyen todas las consultas relevantes. El grupo de consultas y/o los filtros deben ser cambiados para incluir todas las consultas estandard

Resumen de escaneo - FISMA 2014

Category	Description	Issues Found	Best Fix Locations
Access Control*	Organizations must limit information system access to authorized users, processes acting on behalf of authorized users, or devices (including other information systems) and to the types of transactions and functions that authorized users are permitted to exercise.	0	0
Audit And Accountability*	Organizations must: (i) create, protect, and retain information system audit records to the extent needed to enable the monitoring, analysis, investigation, and reporting of unlawful, unauthorized, or inappropriate information system activity; and (ii) ensure that the actions of individual information system users can be uniquely traced to those users so they can be held accountable for their actions.	0	0
Configuration Management*	Organizations must: (i) establish and maintain baseline configurations and inventories of organizational information systems (including hardware, software, firmware, and documentation) throughout the respective system development life cycles; and (ii) establish and enforce security configuration settings for information technology products employed in organizational information systems.	0	0
Identification And Authentication*	Organizations must identify information system users, processes acting on behalf of users, or devices and authenticate (or verify) the identities of those users, processes, or devices, as a prerequisite to allowing access to organizational information systems.	0	0
Media Protection*	Organizations must: (i) protect information system media, both paper and digital; (ii) limit access to information on information system media to authorized users; and (iii) sanitize or destroy information system media before disposal or release for reuse.	12	4
System And Communications Protection*	Organizations must: (i) monitor, control, and protect organizational communications (i.e., information transmitted or received by organizational information systems) at the external boundaries and key internal boundaries of the information systems; and (ii) employ architectural designs, software development techniques, and systems engineering principles that promote effective information security within organizational information systems.	0	0
System And Information Integrity*	Organizations must: (i) identify, report, and correct information and information system flaws in a timely manner; (ii) provide protection from malicious code at appropriate locations within organizational information systems; and (iii) monitor information system security alerts and advisories and take appropriate actions in response.	0	0

***** Los resultados del escaneo del proyecto no incluyen todas las consultas relevantes. El grupo de consultas y/o los filtros deben ser cambiados

para incluir todas las consultas estandard

Resumen de escaneo - NIST SP 800-53

Category	Issues Found	Best Fix Locations
AC-12 Session Termination (P2)	0	0
AC-3 Access Enforcement (P1)*	0	0
AC-4 Information Flow Enforcement (P1)	0	0
AC-6 Least Privilege (P1)	0	0
AU-9 Protection of Audit Information (P1)*	0	0
CM-6 Configuration Settings (P2)	0	0
IA-5 Authenticator Management (P1)	0	0
IA-6 Authenticator Feedback (P2)	0	0
IA-8 Identification and Authentication (Non-Organizational Users) (P1)	0	0
SC-12 Cryptographic Key Establishment and Management (P1)	0	0
SC-13 Cryptographic Protection (P1)	0	0
SC-17 Public Key Infrastructure Certificates (P1)	0	0
SC-18 Mobile Code (P2)	0	0
SC-23 Session Authenticity (P1)*	0	0
SC-28 Protection of Information at Rest (P1)*	0	0
SC-4 Information in Shared Resources (P1)*	0	0
SC-5 Denial of Service Protection (P1)*	2	2
SC-8 Transmission Confidentiality and Integrity (P1)*	12	4
SI-10 Information Input Validation (P1)*	0	0
SI-11 Error Handling (P2)*	0	0
SI-15 Information Output Filtering (P0)*	0	0
SI-16 Memory Protection (P1)*	0	0

***** Los resultados del escaneo del proyecto no incluyen todas las consultas relevantes. El grupo de consultas y/o los filtros deben ser cambiados para incluir todas las consultas estandard

Resumen de escaneo - OWASP Mobile Top 10 2016

Category	Description	Issues Found	Best Fix Locations
M1-Improper Platform Usage*	This category covers misuse of a platform feature or failure to use platform security controls. It might include Android intents, platform permissions, misuse of TouchID, the Keychain, or some other security control that is part of the mobile operating system. There are several ways that mobile apps can experience this risk.	0	0
M2-Insecure Data Storage*	This category covers insecure data storage and unintended data leakage.	0	0
M3-Insecure Communication	This category covers poor handshaking, incorrect SSL versions, weak negotiation, cleartext communication of sensitive assets, etc.	0	0
M4-Insecure Authentication	This category captures notions of authenticating the end user or bad session management. This can include: -Failing to identify the user at all when that should be required -Failure to maintain the user's identity when it is required -Weaknesses in session management	0	0
M5-Insufficient Cryptography*	The code applies cryptography to a sensitive information asset. However, the cryptography is insufficient in some way. Note that anything and everything related to TLS or SSL goes in M3. Also, if the app fails to use cryptography at all when it should, that probably belongs in M2. This category is for issues where cryptography was attempted, but it wasnt done correctly.	0	0
M6-Insecure Authorization	This is a category to capture any failures in authorization (e.g., authorization decisions in the client side, forced browsing, etc.). It is distinct from authentication issues (e.g., device enrolment, user identification, etc.). If the app does not authenticate users at all in a situation where it should (e.g., granting anonymous access to some resource or service when authenticated and authorized access is required), then that is an authentication failure not an authorization failure.	0	0
M7-Client Code Quality*	This category is the catch-all for code-level implementation problems in the mobile client. That's distinct from server-side coding mistakes. This would capture things like buffer overflows, format string vulnerabilities, and various other code-level mistakes where the solution is to rewrite some code that's running on the mobile device.	0	0
M8-Code Tampering	This category covers binary patching, local resource modification, method hooking, method swizzling, and dynamic memory modification. Once the application is delivered to the mobile device, the code and data resources are resident there. An attacker can either directly modify the code, change the contents of memory dynamically, change or replace the system APIs that the application uses, or	0	0

	modify the application's data and resources. This can provide the attacker a direct method of subverting the intended use of the software for personal or monetary gain.		
M9-Reverse Engineering*	This category includes analysis of the final core binary to determine its source code, libraries, algorithms, and other assets. Software such as IDA Pro, Hopper, otool, and other binary inspection tools give the attacker insight into the inner workings of the application. This may be used to exploit other nascent vulnerabilities in the application, as well as revealing information about back end servers, cryptographic constants and ciphers, and intellectual property.	0	0
M10-Extraneous Functionality	Often, developers include hidden backdoor functionality or other internal development security controls that are not intended to be released into a production environment. For example, a developer may accidentally include a password as a comment in a hybrid app. Another example includes disabling of 2-factor authentication during testing.	0	0

***** Los resultados del escaneo del proyecto no incluyen todas las consultas relevantes. El grupo de consultas y/o los filtros deben ser cambiados para incluir todas las consultas estandard

Resumen de escaneo - OWASP Top 10 API

Category	Issues Found	Best Fix Locations
API1-Broken Object Level Authorization	0	0
API2-Broken Authentication	0	0
API3-Excessive Data Exposure	0	0
API4-Lack of Resources and Rate Limiting	0	0
API5-Broken Function Level Authorization	0	0
API6-Mass Assignment	0	0
API7-Security Misconfiguration	0	0
API8-Injection	0	0
API9-Improper Assets Management	0	0
API10-Insufficient Logging and Monitoring	0	0

Resumen de escaneo - Custom

Category	Issues Found	Best Fix Locations
Must audit	0	0
Check	0	0
Optional*	0	0

* Los resultados del escaneo del proyecto no incluyen todas las consultas relevantes. El grupo de consultas y/o los filtros deben ser cambiados para incluir todas las consultas estandard

Resumen de escaneo - ASD STIG 4.10

Category	Issues Found	Best Fix Locations
APSC-DV-000640 - CAT II The application must provide audit record generation capability for the renewal of session IDs.	0	0
APSC-DV-000650 - CAT II The application must not write sensitive data into the application logs.	0	0
APSC-DV-000660 - CAT II The application must provide audit record generation capability for session timeouts.	0	0
APSC-DV-000670 - CAT II The application must record a time stamp indicating when the event occurred.	0	0
APSC-DV-000680 - CAT II The application must provide audit record generation capability for HTTP headers including User-Agent, Referer, GET, and POST.	0	0
APSC-DV-000690 - CAT II The application must provide audit record generation capability for connecting system IP addresses.	0	0
APSC-DV-000700 - CAT II The application must record the username or user ID of the user associated with the event.	0	0
APSC-DV-000710 - CAT II The application must generate audit records when successful/unsuccessful attempts to grant privileges occur.	0	0
APSC-DV-000720 - CAT II The application must generate audit records when successful/unsuccessful attempts to access security objects occur.	0	0
APSC-DV-000730 - CAT II The application must generate audit records when successful/unsuccessful attempts to access security levels occur.	0	0
APSC-DV-000740 - CAT II The application must generate audit records when successful/unsuccessful attempts to access categories of information (e.g., classification levels) occur.	0	0
APSC-DV-000750 - CAT II The application must generate audit records when successful/unsuccessful attempts to modify privileges occur.	0	0
APSC-DV-000760 - CAT II The application must generate audit records when successful/unsuccessful attempts to modify security objects occur.	0	0
APSC-DV-000770 - CAT II The application must generate audit records when successful/unsuccessful attempts to modify security levels occur.	0	0
APSC-DV-000780 - CAT II The application must generate audit records when successful/unsuccessful attempts to modify categories of information (e.g., classification levels) occur.	0	0
APSC-DV-000790 - CAT II The application must generate audit records when successful/unsuccessful attempts to delete privileges occur.	0	0
APSC-DV-000800 - CAT II The application must generate audit records when successful/unsuccessful attempts to delete security levels occur.	0	0
APSC-DV-000810 - CAT II The application must generate audit records when successful/unsuccessful attempts to delete application database security objects occur.	0	0
APSC-DV-000820 - CAT II The application must generate audit records when successful/unsuccessful attempts to delete categories of information (e.g., classification levels) occur.	0	0
APSC-DV-000830 - CAT II The application must generate audit records when successful/unsuccessful logon attempts occur.	0	0
APSC-DV-000840 - CAT II The application must generate audit records for privileged activities or other system-level access.	0	0
APSC-DV-000850 - CAT II The application must generate audit records showing starting and ending time for user access to the system.	0	0
APSC-DV-000860 - CAT II The application must generate audit records when successful/unsuccessful accesses to objects occur.	0	0

APSC-DV-000870 - CAT II The application must generate audit records for all direct access to the information system.	0	0
APSC-DV-000880 - CAT II The application must generate audit records for all account creations, modifications, disabling, and termination events.	0	0
APSC-DV-000910 - CAT II The application must initiate session auditing upon startup.	0	0
APSC-DV-000940 - CAT II The application must log application shutdown events.	0	0
APSC-DV-000950 - CAT II The application must log destination IP addresses.	0	0
APSC-DV-000960 - CAT II The application must log user actions involving access to data.	0	0
APSC-DV-000970 - CAT II The application must log user actions involving changes to data.	0	0
APSC-DV-000980 - CAT II The application must produce audit records containing information to establish when (date and time) the events occurred.	0	0
APSC-DV-000990 - CAT II The application must produce audit records containing enough information to establish which component, feature or function of the application triggered the audit event.	0	0
APSC-DV-001000 - CAT II When using centralized logging; the application must include a unique identifier in order to distinguish itself from other application logs.	0	0
APSC-DV-001010 - CAT II The application must produce audit records that contain information to establish the outcome of the events.	0	0
APSC-DV-001020 - CAT II The application must generate audit records containing information that establishes the identity of any individual or process associated with the event.	0	0
APSC-DV-001030 - CAT II The application must generate audit records containing the full-text recording of privileged commands or the individual identities of group account users.	0	0
APSC-DV-001040 - CAT II The application must implement transaction recovery logs when transaction based.	0	0
APSC-DV-001050 - CAT II The application must provide centralized management and configuration of the content to be captured in audit records generated by all application components.	0	0
APSC-DV-001070 - CAT II The application must off-load audit records onto a different system or media than the system being audited.	0	0
APSC-DV-001080 - CAT II The application must be configured to write application logs to a centralized log repository.	0	0
APSC-DV-001090 - CAT II The application must provide an immediate warning to the SA and ISSO (at a minimum) when allocated audit record storage volume reaches 75% of repository maximum audit record storage capacity.	0	0
APSC-DV-001100 - CAT II Applications categorized as having a moderate or high impact must provide an immediate real-time alert to the SA and ISSO (at a minimum) for all audit failure events.	0	0
APSC-DV-001110 - CAT II The application must alert the ISSO and SA (at a minimum) in the event of an audit processing failure.	0	0
APSC-DV-001120 - CAT II The application must shut down by default upon audit failure (unless availability is an overriding concern).	0	0
APSC-DV-001130 - CAT II The application must provide the capability to centrally review and analyze audit records from multiple components within the system.	0	0
APSC-DV-001140 - CAT II The application must provide the capability to filter audit records for events of interest based upon organization-defined criteria.	0	0
APSC-DV-001150 - CAT II The application must provide an audit reduction capability that supports on-demand reporting requirements.	0	0
APSC-DV-001160 - CAT II The application must provide an audit reduction capability that supports on-demand audit review and analysis.	0	0
APSC-DV-001170 - CAT II The application must provide an audit reduction capability that supports after-the-fact investigations of security incidents.	0	0
APSC-DV-001180 - CAT II The application must provide a report generation capability that supports on-demand audit review and analysis.	0	0

APSC-DV-001190 - CAT II The application must provide a report generation capability that supports on-demand reporting requirements.	0	0
APSC-DV-001200 - CAT II The application must provide a report generation capability that supports after-the-fact investigations of security incidents.	0	0
APSC-DV-001210 - CAT II The application must provide an audit reduction capability that does not alter original content or time ordering of audit records.	0	0
APSC-DV-001220 - CAT II The application must provide a report generation capability that does not alter original content or time ordering of audit records.	0	0
APSC-DV-001250 - CAT II The applications must use internal system clocks to generate time stamps for audit records.	0	0
APSC-DV-001260 - CAT II The application must record time stamps for audit records that can be mapped to Coordinated Universal Time (UTC) or Greenwich Mean Time (GMT).	0	0
APSC-DV-001270 - CAT II The application must record time stamps for audit records that meet a granularity of one second for a minimum degree of precision.	0	0
APSC-DV-001280 - CAT II The application must protect audit information from any type of unauthorized read access.	0	0
APSC-DV-001290 - CAT II The application must protect audit information from unauthorized modification.	0	0
APSC-DV-001300 - CAT II The application must protect audit information from unauthorized deletion.	0	0
APSC-DV-001310 - CAT II The application must protect audit tools from unauthorized access.	0	0
APSC-DV-001320 - CAT II The application must protect audit tools from unauthorized modification.	0	0
APSC-DV-001330 - CAT II The application must protect audit tools from unauthorized deletion.	0	0
APSC-DV-001340 - CAT II The application must back up audit records at least every seven days onto a different system or system component than the system or component being audited.	0	0
APSC-DV-001570 - CAT II The application must electronically verify Personal Identity Verification (PIV) credentials.	0	0
APSC-DV-001350 - CAT II The application must use cryptographic mechanisms to protect the integrity of audit information.	0	0
APSC-DV-001360 - CAT II Application audit tools must be cryptographically hashed.	0	0
APSC-DV-001370 - CAT II The integrity of the audit tools must be validated by checking the files for changes in the cryptographic hash value.	0	0
APSC-DV-001390 - CAT II The application must prohibit user installation of software without explicit privileged status.	0	0
APSC-DV-001410 - CAT II The application must enforce access restrictions associated with changes to application configuration.	0	0
APSC-DV-001420 - CAT II The application must audit who makes configuration changes to the application.	0	0
APSC-DV-001430 - CAT II The application must have the capability to prevent the installation of patches, service packs, or application components without verification the software component has been digitally signed using a certificate that is recognized and approved by the orga	0	0
APSC-DV-001440 - CAT II The applications must limit privileges to change the software resident within software libraries.	0	0
APSC-DV-001460 - CAT II An application vulnerability assessment must be conducted.	0	0
APSC-DV-001480 - CAT II The application must prevent program execution in accordance with organization-defined policies regarding software program usage and restrictions, and/or rules authorizing the terms and conditions of software program usage.	0	0
APSC-DV-001490 - CAT II The application must employ a deny-all, permit-by-exception (whitelist) policy to allow the execution of authorized software programs.	0	0
APSC-DV-001500 - CAT II The application must be configured to disable non-essential capabilities.	0	0
APSC-DV-001510 - CAT II The application must be configured to use only functions, ports, and protocols permitted to it in the PPSM CAL.	0	0

APSC-DV-001520 - CAT II The application must require users to reauthenticate when organization-defined circumstances or situations require reauthentication.	0	0
APSC-DV-001530 - CAT II The application must require devices to reauthenticate when organization-defined circumstances or situations requiring reauthentication.	0	0
APSC-DV-001540 - CAT I The application must uniquely identify and authenticate organizational users (or processes acting on behalf of organizational users).	0	0
APSC-DV-001550 - CAT II The application must use multifactor (Alt. Token) authentication for network access to privileged accounts.	0	0
APSC-DV-001560 - CAT II The application must accept Personal Identity Verification (PIV) credentials.	0	0
APSC-DV-001580 - CAT II The application must use multifactor (e.g., CAC, Alt. Token) authentication for network access to non-privileged accounts.	0	0
APSC-DV-001590 - CAT II The application must use multifactor (Alt. Token) authentication for local access to privileged accounts.	0	0
APSC-DV-001600 - CAT II The application must use multifactor (e.g., CAC, Alt. Token) authentication for local access to non-privileged accounts.	0	0
APSC-DV-001610 - CAT II The application must ensure users are authenticated with an individual authenticator prior to using a group authenticator.	0	0
APSC-DV-001620 - CAT II The application must implement replay-resistant authentication mechanisms for network access to privileged accounts.	0	0
APSC-DV-001630 - CAT II The application must implement replay-resistant authentication mechanisms for network access to non-privileged accounts.	0	0
APSC-DV-001640 - CAT II The application must utilize mutual authentication when endpoint device non-repudiation protections are required by DoD policy or by the data owner.	0	0
APSC-DV-001650 - CAT II The application must authenticate all network connected endpoint devices before establishing any connection.	0	0
APSC-DV-001660 - CAT II Service-Oriented Applications handling non-releasable data must authenticate endpoint devices via mutual SSL/TLS.	0	0
APSC-DV-001670 - CAT II The application must disable device identifiers after 35 days of inactivity unless a cryptographic certificate is used for authentication.	0	0
APSC-DV-001680 - CAT I The application must enforce a minimum 15-character password length.	0	0
APSC-DV-001690 - CAT II The application must enforce password complexity by requiring that at least one upper-case character be used.	0	0
APSC-DV-001700 - CAT II The application must enforce password complexity by requiring that at least one lower-case character be used.	0	0
APSC-DV-001710 - CAT II The application must enforce password complexity by requiring that at least one numeric character be used.	0	0
APSC-DV-001720 - CAT II The application must enforce password complexity by requiring that at least one special character be used.	0	0
APSC-DV-001730 - CAT II The application must require the change of at least 8 of the total number of characters when passwords are changed.	0	0
APSC-DV-001740 - CAT I The application must only store cryptographic representations of passwords.	12	4
APSC-DV-001850 - CAT I The application must not display passwords/PINs as clear text.	0	0
APSC-DV-001750 - CAT I The application must transmit only cryptographically-protected passwords.	0	0
APSC-DV-001760 - CAT II The application must enforce 24 hours/1 day as the minimum password lifetime.	0	0
APSC-DV-001770 - CAT II The application must enforce a 60-day maximum password lifetime restriction.	0	0
APSC-DV-001780 - CAT II The application must prohibit password reuse for a minimum of five generations.	0	0
APSC-DV-001790 - CAT II The application must allow the use of a temporary password for system logons with an immediate change to a permanent password.	0	0

APSC-DV-001795 - CAT II The application password must not be changeable by users other than the administrator or the user with which the password is associated.	0	0
APSC-DV-001800 - CAT II The application must terminate existing user sessions upon account deletion.	0	0
APSC-DV-001820 - CAT I The application, when using PKI-based authentication, must enforce authorized access to the corresponding private key.	0	0
APSC-DV-001830 - CAT II The application must map the authenticated identity to the individual user or group account for PKI-based authentication.	0	0
APSC-DV-001870 - CAT II The application must uniquely identify and authenticate non-organizational users (or processes acting on behalf of non-organizational users).	0	0
APSC-DV-001810 - CAT I The application, when utilizing PKI-based authentication, must validate certificates by constructing a certification path (which includes status information) to an accepted trust anchor.	0	0
APSC-DV-001840 - CAT II The application, for PKI-based authentication, must implement a local cache of revocation data to support path discovery and validation in case of the inability to access revocation information via the network.	0	0
APSC-DV-001860 - CAT II The application must use mechanisms meeting the requirements of applicable federal laws, Executive Orders, directives, policies, regulations, standards, and guidance for authentication to a cryptographic module.	0	0
APSC-DV-001880 - CAT II The application must accept Personal Identity Verification (PIV) credentials from other federal agencies.	0	0
APSC-DV-001890 - CAT II The application must electronically verify Personal Identity Verification (PIV) credentials from other federal agencies.	0	0
APSC-DV-002050 - CAT II Applications making SAML assertions must use FIPS-approved random numbers in the generation of SessionIndex in the SAML element AuthnStatement.	0	0
APSC-DV-001900 - CAT II The application must accept FICAM-approved third-party credentials.	0	0
APSC-DV-001910 - CAT II The application must conform to FICAM-issued profiles.	0	0
APSC-DV-001930 - CAT II Applications used for non-local maintenance sessions must audit non-local maintenance and diagnostic sessions for organization-defined auditable events.	0	0
APSC-DV-000310 - CAT III The application must have a process, feature or function that prevents removal or disabling of emergency accounts.	0	0
APSC-DV-001940 - CAT II Applications used for non-local maintenance sessions must implement cryptographic mechanisms to protect the integrity of non-local maintenance and diagnostic communications.	0	0
APSC-DV-001950 - CAT II Applications used for non-local maintenance sessions must implement cryptographic mechanisms to protect the confidentiality of non-local maintenance and diagnostic communications.	0	0
APSC-DV-001960 - CAT II Applications used for non-local maintenance sessions must verify remote disconnection at the termination of non-local maintenance and diagnostic sessions.	0	0
APSC-DV-001970 - CAT II The application must employ strong authenticators in the establishment of non-local maintenance and diagnostic sessions.	0	0
APSC-DV-001980 - CAT II The application must terminate all sessions and network connections when non-local maintenance is completed.	0	0
APSC-DV-001995 - CAT II The application must not be vulnerable to race conditions.	0	0
APSC-DV-002000 - CAT II The application must terminate all network connections associated with a communications session at the end of the session.	0	0
APSC-DV-002010 - CAT II The application must implement NSA-approved cryptography to protect classified information in accordance with applicable federal laws, Executive Orders, directives, policies, regulations, and standards.	0	0
APSC-DV-002020 - CAT II The application must utilize FIPS-validated cryptographic modules when signing application components.	0	0
APSC-DV-002030 - CAT II The application must utilize FIPS-validated cryptographic modules when generating cryptographic hashes.	0	0

APSC-DV-002040 - CAT II The application must utilize FIPS-validated cryptographic modules when protecting unclassified information that requires cryptographic protection.	0	0
APSC-DV-002150 - CAT II The application user interface must be either physically or logically separated from data storage and management interfaces.	0	0
APSC-DV-002210 - CAT II The application must set the HTTPOnly flag on session cookies.	0	0
APSC-DV-002220 - CAT II The application must set the secure flag on session cookies.	0	0
APSC-DV-002230 - CAT I The application must not expose session IDs.	0	0
APSC-DV-002240 - CAT I The application must destroy the session ID value and/or cookie on logoff or browser close.	0	0
APSC-DV-002250 - CAT II Applications must use system-generated session identifiers that protect against session fixation.	0	0
APSC-DV-002260 - CAT II Applications must validate session identifiers.	0	0
APSC-DV-002270 - CAT II Applications must not use URL embedded session IDs.	0	0
APSC-DV-002280 - CAT II The application must not re-use or recycle session IDs.	0	0
APSC-DV-002290 - CAT II The application must use the Federal Information Processing Standard (FIPS) 140-2-validated cryptographic modules and random number generator if the application implements encryption, key exchange, digital signature, and hash functionality.	0	0
APSC-DV-002300 - CAT II The application must only allow the use of DoD-approved certificate authorities for verification of the establishment of protected sessions.	0	0
APSC-DV-002310 - CAT I The application must fail to a secure state if system initialization fails, shutdown fails, or aborts fail.	0	0
APSC-DV-002320 - CAT II In the event of a system failure, applications must preserve any information necessary to determine cause of failure and any information necessary to return to operations with least disruption to mission processes.	0	0
APSC-DV-002330 - CAT II The application must protect the confidentiality and integrity of stored information when required by DoD policy or the information owner.*	0	0
APSC-DV-002340 - CAT II The application must implement approved cryptographic mechanisms to prevent unauthorized modification of organization-defined information at rest on organization-defined information system components.	0	0
APSC-DV-002350 - CAT II The application must use appropriate cryptography in order to protect stored DoD information when required by the information owner or DoD policy.	0	0
APSC-DV-002360 - CAT II The application must isolate security functions from non-security functions.	0	0
APSC-DV-002370 - CAT II The application must maintain a separate execution domain for each executing process.	0	0
APSC-DV-002380 - CAT II Applications must prevent unauthorized and unintended information transfer via shared system resources.	0	0
APSC-DV-002390 - CAT II XML-based applications must mitigate DoS attacks by using XML filters, parser options, or gateways.	0	0
APSC-DV-002400 - CAT II The application must restrict the ability to launch Denial of Service (DoS) attacks against itself or other information systems.*	2	2
APSC-DV-002410 - CAT II The web service design must include redundancy mechanisms when used with high-availability systems.	0	0
APSC-DV-002420 - CAT II An XML firewall function must be deployed to protect web services when exposed to untrusted networks.	0	0
APSC-DV-002610 - CAT II The application must remove organization-defined software components after updated versions have been installed.	0	0
APSC-DV-002440 - CAT I The application must protect the confidentiality and integrity of transmitted information.	0	0
APSC-DV-002450 - CAT II The application must implement cryptographic mechanisms to prevent unauthorized disclosure of information and/or detect changes to information during transmission unless otherwise protected by alternative physical safeguards, such as, at a minimum, a Prot	0	0

APSC-DV-002460 - CAT II The application must maintain the confidentiality and integrity of information during preparation for transmission.	0	0
APSC-DV-002470 - CAT II The application must maintain the confidentiality and integrity of information during reception.	0	0
APSC-DV-002480 - CAT II The application must not disclose unnecessary information to users.	0	0
APSC-DV-002485 - CAT I The application must not store sensitive information in hidden fields.	0	0
APSC-DV-002490 - CAT I The application must protect from Cross-Site Scripting (XSS) vulnerabilities.	0	0
APSC-DV-002500 - CAT II The application must protect from Cross-Site Request Forgery (CSRF) vulnerabilities.*	0	0
APSC-DV-002510 - CAT I The application must protect from command injection.	0	0
APSC-DV-002520 - CAT II The application must protect from canonical representation vulnerabilities.	0	0
APSC-DV-002530 - CAT II The application must validate all input.*	0	0
APSC-DV-002540 - CAT I The application must not be vulnerable to SQL Injection.	0	0
APSC-DV-002550 - CAT I The application must not be vulnerable to XML-oriented attacks.	0	0
APSC-DV-002560 - CAT I The application must not be subject to input handling vulnerabilities.*	0	0
APSC-DV-002570 - CAT II The application must generate error messages that provide information necessary for corrective actions without revealing information that could be exploited by adversaries.	0	0
APSC-DV-002580 - CAT II The application must reveal error messages only to the ISSO, ISSM, or SA.*	0	0
APSC-DV-002590 - CAT I The application must not be vulnerable to overflow attacks.*	0	0
APSC-DV-002630 - CAT II Security-relevant software updates and patches must be kept up to date.	0	0
APSC-DV-002760 - CAT II The application performing organization-defined security functions must verify correct operation of security functions.	0	0
APSC-DV-002900 - CAT II The ISSO must ensure application audit trails are retained for at least 1 year for applications without SAMI data, and 5 years for applications including SAMI data.	0	0
APSC-DV-002770 - CAT II The application must perform verification of the correct operation of security functions: upon system startup and/or restart; upon command by a user with privileged access; and/or every 30 days.	0	0
APSC-DV-002780 - CAT III The application must notify the ISSO and ISSM of failed security verification tests.	0	0
APSC-DV-002870 - CAT II Unsigned Category 1A mobile code must not be used in the application in accordance with DoD policy.	0	0
APSC-DV-002880 - CAT II The ISSO must ensure an account management process is implemented, verifying only authorized users can gain access to the application, and individual accounts designated as inactive, suspended, or terminated are promptly removed.	0	0
APSC-DV-002890 - CAT I Application web servers must be on a separate network segment from the application and database servers if it is a tiered application operating in the DoD DMZ.	0	0
APSC-DV-002910 - CAT II The ISSO must review audit trails periodically based on system documentation recommendations or immediately upon system security events.	0	0
APSC-DV-002920 - CAT II The ISSO must report all suspected violations of IA policies in accordance with DoD information system IA procedures.	0	0
APSC-DV-002930 - CAT II The ISSO must ensure active vulnerability testing is performed.	0	0
APSC-DV-002980 - CAT II New IP addresses, data services, and associated ports used by the application must be submitted to the appropriate approving authority for the organization, which in turn will be submitted through the DoD Ports, Protocols, and Services Management (DoD PPS	0	0
APSC-DV-002950 - CAT II Execution flow diagrams and design documents must be created to show how deadlock and recursion issues in web services are being mitigated.	0	0
APSC-DV-002960 - CAT II The designer must ensure the application does not store configuration and control files in the same directory as user data.	0	0
APSC-DV-002970 - CAT II The ISSO must ensure if a DoD STIG or NSA guide is not available, a third-party	0	0

product will be configured by following available guidance.		
APSC-DV-002990 - CAT II The application must be registered with the DoD Ports and Protocols Database.	0	0
APSC-DV-002990 - CAT II The application must be registered with the DoD Ports and Protocols Database.	0	0
APSC-DV-002995 - CAT II The Configuration Management (CM) repository must be properly patched and STIG compliant.	0	0
APSC-DV-003000 - CAT II Access privileges to the Configuration Management (CM) repository must be reviewed every three months.	0	0
APSC-DV-003010 - CAT II A Software Configuration Management (SCM) plan describing the configuration control and change management process of application objects developed by the organization and the roles and responsibilities of the organization must be created and maintained.	0	0
APSC-DV-003020 - CAT II A Configuration Control Board (CCB) that meets at least every release cycle, for managing the Configuration Management (CM) process must be established.	0	0
APSC-DV-003030 - CAT II The application services and interfaces must be compatible with and ready for IPv6 networks.	0	0
APSC-DV-003040 - CAT II The application must not be hosted on a general purpose machine if the application is designated as critical or high availability by the ISSO.	0	0
APSC-DV-003050 - CAT II A disaster recovery/continuity plan must exist in accordance with DoD policy based on the applications availability requirements.	0	0
APSC-DV-003060 - CAT II Recovery procedures and technical system features must exist so recovery is performed in a secure and verifiable manner. The ISSO will document circumstances inhibiting a trusted recovery.	0	0
APSC-DV-003070 - CAT II Data backup must be performed at required intervals in accordance with DoD policy.	0	0
APSC-DV-003080 - CAT II Back-up copies of the application software or source code must be stored in a fire-rated container or stored separately (offsite).	0	0
APSC-DV-003090 - CAT II Procedures must be in place to assure the appropriate physical and technical protection of the backup and restoration of the application.	0	0
APSC-DV-003100 - CAT II The application must use encryption to implement key exchange and authenticate endpoints prior to establishing a communication channel for key exchange.	0	0
APSC-DV-003110 - CAT I The application must not contain embedded authentication data.	0	0
APSC-DV-003120 - CAT I The application must have the capability to mark sensitive/classified output when required.	0	0
APSC-DV-003130 - CAT III Prior to each release of the application, updates to system, or applying patches; tests plans and procedures must be created and executed.	0	0
APSC-DV-003150 - CAT II At least one tester must be designated to test for security flaws in addition to functional testing.	0	0
APSC-DV-003140 - CAT II Application files must be cryptographically hashed prior to deploying to DoD operational networks.	0	0
APSC-DV-003160 - CAT III Test procedures must be created and at least annually executed to ensure system initialization, shutdown, and aborts are configured to verify the system remains in a secure state.	0	0
APSC-DV-003170 - CAT II An application code review must be performed on the application.	0	0
APSC-DV-003180 - CAT III Code coverage statistics must be maintained for each release of the application.	0	0
APSC-DV-003190 - CAT II Flaws found during a code review must be tracked in a defect tracking system.	0	0
APSC-DV-003200 - CAT II The changes to the application must be assessed for IA and accreditation impact prior to implementation.	0	0
APSC-DV-003210 - CAT II Security flaws must be fixed or addressed in the project plan.	0	0
APSC-DV-003215 - CAT III The application development team must follow a set of coding standards.	0	0
APSC-DV-003220 - CAT III The designer must create and update the Design Document for each release of the application.	0	0

APSC-DV-003230 - CAT II Threat models must be documented and reviewed for each application release and updated as required by design and functionality changes or when new threats are discovered.	0	0
APSC-DV-003235 - CAT II The application must not be subject to error handling vulnerabilities.*	0	0
APSC-DV-003250 - CAT I The application must be decommissioned when maintenance or support is no longer available.	0	0
APSC-DV-003236 - CAT II The application development team must provide an application incident response plan.	0	0
APSC-DV-003240 - CAT I All products must be supported by the vendor or the development team.	0	0
APSC-DV-003260 - CAT III Procedures must be in place to notify users when an application is decommissioned.	0	0
APSC-DV-003270 - CAT II Unnecessary built-in application accounts must be disabled.	0	0
APSC-DV-003280 - CAT I Default passwords must be changed.	0	0
APSC-DV-003330 - CAT II The system must alert an administrator when low resource conditions are encountered.	0	0
APSC-DV-003285 - CAT II An Application Configuration Guide must be created and included with the application.	0	0
APSC-DV-003290 - CAT II If the application contains classified data, a Security Classification Guide must exist containing data elements and their classification.	0	0
APSC-DV-003300 - CAT II The designer must ensure uncategorized or emerging mobile code is not used in applications.	0	0
APSC-DV-003310 - CAT II Production database exports must have database administration credentials and sensitive data removed before releasing the export.	0	0
APSC-DV-003320 - CAT II Protections against DoS attacks must be implemented.	0	0
APSC-DV-003340 - CAT III At least one application administrator must be registered to receive update notifications, or security alerts, when automated alerts are available.	0	0
APSC-DV-003360 - CAT III The application must generate audit records when concurrent logons from different workstations occur.	0	0
APSC-DV-003345 - CAT III The application must provide notifications or alerts when product update and security related patches are available.	0	0
APSC-DV-003350 - CAT II Connections between the DoD enclave and the Internet or other public or commercial wide area networks must require a DMZ.	0	0
APSC-DV-003400 - CAT II The Program Manager must verify all levels of program management, designers, developers, and testers receive annual security training pertaining to their job function.	0	0
APSC-DV-000010 - CAT II The application must provide a capability to limit the number of logon sessions per user.	0	0
APSC-DV-000060 - CAT II The application must clear temporary storage and cookies when the session is terminated.	0	0
APSC-DV-000070 - CAT II The application must automatically terminate the non-privileged user session and log off non-privileged users after a 15 minute idle time period has elapsed.	0	0
APSC-DV-000080 - CAT II The application must automatically terminate the admin user session and log off admin users after a 10 minute idle time period is exceeded.	0	0
APSC-DV-000090 - CAT II Applications requiring user access authentication must provide a logoff capability for user initiated communication session.	0	0
APSC-DV-000100 - CAT III The application must display an explicit logoff message to users indicating the reliable termination of authenticated communications sessions.	0	0
APSC-DV-000110 - CAT II The application must associate organization-defined types of security attributes having organization-defined security attribute values with information in storage.	0	0
APSC-DV-000120 - CAT II The application must associate organization-defined types of security attributes having organization-defined security attribute values with information in process.	0	0
APSC-DV-000130 - CAT II The application must associate organization-defined types of security attributes	0	0

having organization-defined security attribute values with information in transmission.		
APSC-DV-000160 - CAT II The application must implement DoD-approved encryption to protect the confidentiality of remote access sessions.	0	0
APSC-DV-000170 - CAT II The application must implement cryptographic mechanisms to protect the integrity of remote access sessions.	0	0
APSC-DV-000190 - CAT I Messages protected with WS_Security must use time stamps with creation and expiration times.	0	0
APSC-DV-000180 - CAT II Applications with SOAP messages requiring integrity must include the following message elements:-Message ID-Service Request-Timestamp-SAML Assertion (optionally included in messages) and all elements of the message must be digitally signed.	0	0
APSC-DV-000200 - CAT I Validity periods must be verified on all application messages using WS-Security or SAML assertions.	0	0
APSC-DV-000210 - CAT II The application must ensure each unique asserting party provides unique assertion ID references for each SAML assertion.	0	0
APSC-DV-000220 - CAT II The application must ensure encrypted assertions, or equivalent confidentiality protections are used when assertion data is passed through an intermediary, and confidentiality of the assertion data is required when passing through the intermediary.	0	0
APSC-DV-000230 - CAT I The application must use the NotOnOrAfter condition when using the SubjectConfirmation element in a SAML assertion.	0	0
APSC-DV-000240 - CAT I The application must use both the NotBefore and NotOnOrAfter elements or OneTimeUse element when using the Conditions element in a SAML assertion.	0	0
APSC-DV-000250 - CAT II The application must ensure if a OneTimeUse element is used in an assertion, there is only one of the same used in the Conditions element portion of an assertion.	0	0
APSC-DV-000260 - CAT II The application must ensure messages are encrypted when the SessionIndex is tied to privacy data.	0	0
APSC-DV-000290 - CAT II Shared/group account credentials must be terminated when members leave the group.	0	0
APSC-DV-000280 - CAT II The application must provide automated mechanisms for supporting account management functions.	0	0
APSC-DV-000300 - CAT II The application must automatically remove or disable temporary user accounts 72 hours after account creation.	0	0
APSC-DV-000320 - CAT III The application must automatically disable accounts after a 35 day period of account inactivity.	0	0
APSC-DV-000330 - CAT II Unnecessary application accounts must be disabled, or deleted.	0	0
APSC-DV-000420 - CAT II The application must automatically audit account enabling actions.	0	0
APSC-DV-000340 - CAT II The application must automatically audit account creation.	0	0
APSC-DV-000350 - CAT II The application must automatically audit account modification.	0	0
APSC-DV-000360 - CAT II The application must automatically audit account disabling actions.	0	0
APSC-DV-000370 - CAT II The application must automatically audit account removal actions.	0	0
APSC-DV-000380 - CAT III The application must notify System Administrators and Information System Security Officers when accounts are created.	0	0
APSC-DV-000390 - CAT III The application must notify System Administrators and Information System Security Officers when accounts are modified.	0	0
APSC-DV-000400 - CAT III The application must notify System Administrators and Information System Security Officers of account disabling actions.	0	0
APSC-DV-000410 - CAT III The application must notify System Administrators and Information System Security Officers of account removal actions.	0	0
APSC-DV-000430 - CAT III The application must notify System Administrators and Information System Security Officers of account enabling actions.	0	0
APSC-DV-000440 - CAT II Application data protection requirements must be identified and documented.	0	0

APSC-DV-000520 - CAT II The application must audit the execution of privileged functions.	0	0
APSC-DV-000450 - CAT II The application must utilize organization-defined data mining detection techniques for organization-defined data storage objects to adequately detect data mining attempts.	0	0
APSC-DV-000460 - CAT I The application must enforce approved authorizations for logical access to information and system resources in accordance with applicable access control policies.	0	0
APSC-DV-000470 - CAT II The application must enforce organization-defined discretionary access control policies over defined subjects and objects.	0	0
APSC-DV-000480 - CAT II The application must enforce approved authorizations for controlling the flow of information within the system based on organization-defined information flow control policies.	0	0
APSC-DV-000490 - CAT II The application must enforce approved authorizations for controlling the flow of information between interconnected systems based on organization-defined information flow control policies.	0	0
APSC-DV-000500 - CAT II The application must prevent non-privileged users from executing privileged functions to include disabling, circumventing, or altering implemented security safeguards/countermeasures.	0	0
APSC-DV-000510 - CAT I The application must execute without excessive account permissions.	0	0
APSC-DV-000530 - CAT I The application must enforce the limit of three consecutive invalid logon attempts by a user during a 15 minute time period.	0	0
APSC-DV-000560 - CAT III The application must retain the Standard Mandatory DoD Notice and Consent Banner on the screen until users acknowledge the usage conditions and take explicit actions to log on for further access.	0	0
APSC-DV-000540 - CAT II The application administrator must follow an approved process to unlock locked user accounts.	0	0
APSC-DV-000550 - CAT III The application must display the Standard Mandatory DoD Notice and Consent Banner before granting access to the application.	0	0
APSC-DV-000570 - CAT III The publicly accessible application must display the Standard Mandatory DoD Notice and Consent Banner before granting access to the application.	0	0
APSC-DV-000580 - CAT III The application must display the time and date of the users last successful logon.	0	0
APSC-DV-000630 - CAT II The application must provide audit record generation capability for the destruction of session IDs.	0	0
APSC-DV-000590 - CAT II The application must protect against an individual (or process acting on behalf of an individual) falsely denying having performed organization-defined actions to be covered by non-repudiation.	0	0
APSC-DV-000600 - CAT II For applications providing audit record aggregation, the application must compile audit records from organization-defined information system components into a system-wide audit trail that is time-correlated with an organization-defined level of tolerance	0	0
APSC-DV-000610 - CAT II The application must provide the capability for organization-identified individuals or roles to change the auditing to be performed on all application components, based on all selectable event criteria within organization-defined time thresholds.	0	0
APSC-DV-000620 - CAT II The application must provide audit record generation capability for the creation of session IDs.	0	0

***** Los resultados del escaneo del proyecto no incluyen todas las consultas relevantes. El grupo de consultas y/o los filtros deben ser cambiados para incluir todas las consultas estandard

Resumen de escaneo - OWASP Top 10 2010

Category	Issues Found	Best Fix Locations
A1-Injection	0	0
A2-Cross-Site Scripting (XSS)	0	0
A3-Broken Authentication and Session Management	0	0
A4-Insecure Direct Object References	0	0
A5-Cross-Site Request Forgery (CSRF)	0	0
A6-Security Misconfiguration	0	0
A7-Insecure Cryptographic Storage	0	0
A8-Failure to Restrict URL Access	0	0
A9-Insufficient Transport Layer Protection	0	0
A10-Unvalidated Redirects and Forwards	0	0

Resumen de escaneo - MOIS(KISA) Secure Coding 2021

Category	Issues Found	Best Fix Locations
MOIS(KISA) API misuse*	1	1
MOIS(KISA) Code error	1	1
MOIS(KISA) Encapsulation	0	0
MOIS(KISA) Error processing*	0	0
MOIS(KISA) Security Functions*	12	4
MOIS(KISA) Time and status	0	0
MOIS(KISA) Verification and representation of input data*	0	0

***** Los resultados del escaneo del proyecto no incluyen todas las consultas relevantes. El grupo de consultas y/o los filtros deben ser cambiados para incluir todas las consultas estandard

Resumen de escaneo - SANS top 25

Category	Issues Found	Best Fix Locations
SANS top 25*	13	5

***** Los resultados del escaneo del proyecto no incluyen todas las consultas relevantes. El grupo de consultas y/o los filtros deben ser cambiados para incluir todas las consultas estandard

Resumen de escaneo - CWE top 25

Category	Issues Found	Best Fix Locations
CWE top 25*	13	5

***** Los resultados del escaneo del proyecto no incluyen todas las consultas relevantes. El grupo de consultas y/o los filtros deben ser cambiados para incluir todas las consultas estandard

Resumen de escaneo - OWASP ASVS

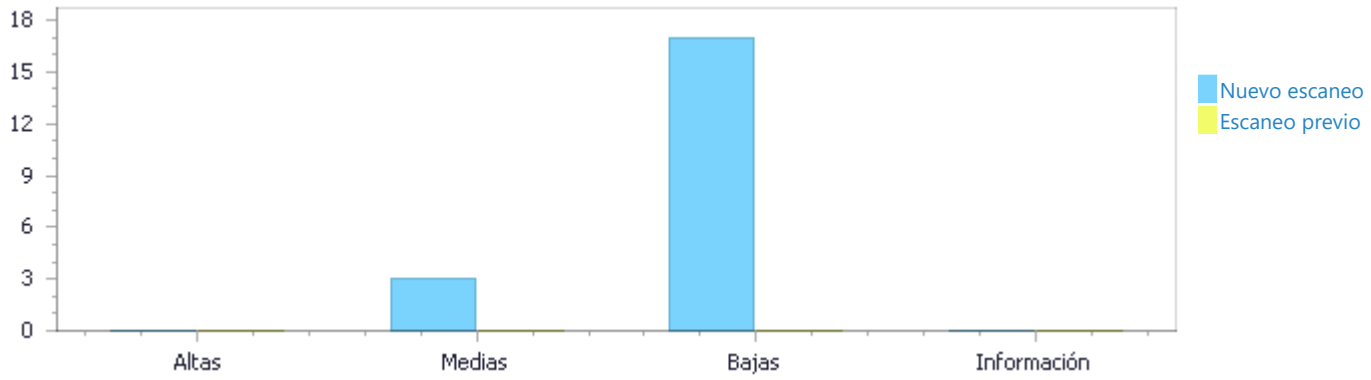
Category	Issues Found	Best Fix Locations
V01 Architecture, Design and Threat Modeling*	0	0
V02 Authentication*	12	4
V03 Session Management*	0	0
V04 Access Control	0	0
V05 Validation, Sanitization and Encoding*	0	0
V06 Stored Cryptography*	0	0
V07 Error Handling and Logging*	0	0
V08 Data Protection*	0	0
V09 Communication*	0	0
V10 Malicious Code*	0	0
V11 Business Logic*	0	0
V12 Files and Resources*	0	0
V13 API and Web Service*	0	0
V14 Configuration*	0	0

***** Los resultados del escaneo del proyecto no incluyen todas las consultas relevantes. El grupo de consultas y/o los filtros deben ser cambiados para incluir todas las consultas estandard

Distribución de resultados por estatus

Primer escaneo del proyecto

	Altas	Medias	Bajas	Información	Total
Nuevas vulnerabilidades	0	3	17	0	20
Vulnerabilidades recurrentes	0	0	0	0	0
Total	0	3	17	0	20
Vulnerabilidades solucionadas	0	0	0	0	0



Distribución de resultados por estado

	Altas	Medias	Bajas	Información	Total
Para verificar	0	3	17	0	20
No explotable	0	0	0	0	0
Confirmado	0	0	0	0	0
Urgente	0	0	0	0	0
Propuesto no explotable	0	0	0	0	0
Total	0	3	17	0	20

Resumen de los resultados

Tipo de vulnerabilidad	Ocurrencias	Severidad

Dangerous Functions	1	Medias
Memory Leak	1	Medias
Use of Uninitialized Variable	1	Medias
Insufficiently Protected Credentials	12	Bajas
Use of Sizeof On a Pointer Type	5	Bajas

Los 10 archivos más vulnerables

Vulnerabilidades altas y medias

Nombre del archivo	Problemas encontrados
VALIDADORAPORTACIONESVOLUNTARIAS 13122022/Clases/C_dir.cpp	1
VALIDADORAPORTACIONESVOLUNTARIAS 13122022/dlgValidacionIfes.cpp	1
VALIDADORAPORTACIONESVOLUNTARIAS 13122022/DlgImagFaltante.h	1
VALIDADORAPORTACIONESVOLUNTARIAS 13122022/DlgImagFaltante.cpp	1

Detalles de los Resultados del escaneo

Dangerous Functions

Ruta de consulta:
CPP\Cx\CPP Medium Threat\Dangerous Functions Versión:2

Categorías

OWASP Top 10 2013: A9-Using Components with Known Vulnerabilities
OWASP Top 10 2017: A9-Using Components with Known Vulnerabilities
OWASP Top 10 2021: A4-Insecure Design
MOIS(KISA) Secure Coding 2021: MOIS(KISA) API misuse

Descripción

Dangerous Functions\Ruta 1:

Severidad	Medias
Estado del resultado	Para verificar
Resultados en línea	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1155844&projectid=45512&pathid=1
Estatus	Nuevo
Detection Date	12/14/2022 10:10:28 AM

	Origen	Destino
Archivo	VALIDADORAPORTACIONESVOLUNTARIAS 13122022/Clases/C_dir.cpp	VALIDADORAPORTACIONESVOLUNTARIAS 13122022/Clases/C_dir.cpp
Línea	37	37
Objeto	memcpy	memcpy

Fragmento de código

Nombre del archivo: VALIDADORAPORTACIONESVOLUNTARIAS 13122022/Clases/C_dir.cpp

Método: int C_Directorio::archivo(struct _finddata_t *nombre)

```
....
37.  memcpy(nombre, &encontro, sizeof(_finddata_t));
```

Memory Leak

Ruta de consulta:
CPP\Cx\CPP Medium Threat\Memory Leak Versión:5

Categorías

NIST SP 800-53: SC-5 Denial of Service Protection (P1)
ASD STIG 4.10: APSC-DV-002400 - CAT II The application must restrict the ability to launch Denial of Service (DoS) attacks against itself or other information systems.

Descripción

Memory Leak\Ruta 1:

Severidad	Medias
Estado del resultado	Para verificar
Resultados en línea	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1155844&projectid=45512&pathid=2
Estatus	Nuevo
Detection Date	12/14/2022 10:10:28 AM

	Origen	Destino
Archivo	VALIDADORAPORTACIONESVOLUNTARIAS 13122022/dlgValidacionIfes.cpp	VALIDADORAPORTACIONESVOLUNTARIAS 13122022/dlgValidacionIfes.cpp
Línea	86	86
Objeto	m_ImgLst	m_ImgLst

Fragmento de código

Nombre del archivo: VALIDADORAPORTACIONESVOLUNTARIAS 13122022/dlgValidacionIfes.cpp

Método: BOOL CdlgValidacionIfes::OnInitDialog()

```
....
86.  m_ImgLst = new CImageList();
```

Use of Uninitialized Variable

Ruta de consulta:

CPP\Cx\CPP Medium Threat\Use of Uninitialized Variable Versión:0

Categorías

NIST SP 800-53: SC-5 Denial of Service Protection (P1)
ASD STIG 4.10: APSC-DV-002400 - CAT II The application must restrict the ability to launch Denial of Service (DoS) attacks against itself or other information systems.
OWASP Top 10 2021: A4-Insecure Design
MOIS(KISA) Secure Coding 2021: MOIS(KISA) Code error
SANS top 25: SANS top 25
CWE top 25: CWE top 25

Descripción

Use of Uninitialized Variable\Ruta 1:

Severidad	Medias
Estado del resultado	Para verificar
Resultados en línea	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1155844&projectid=45512&pathid=3
Estatus	Nuevo
Detection Date	12/14/2022 10:10:29 AM

	Origen	Destino
Archivo	VALIDADORAPORTACIONESVOLUNTARIAS 13122022/DlgImagFaltante.h	VALIDADORAPORTACIONESVOLUNTARIAS 13122022/DlgImagFaltante.cpp
Línea	25	213

Objeto	iTipoRechazo	iTipoRechazo

Fragmento de código

Nombre del archivo VALIDADORAPORTACIONESVOLUNTARIAS 13122022/DlgImagFaltante.h

Método int iTipoRechazo;

```
....
25.  int iTipoRechazo;
```

Nombre del archivo VALIDADORAPORTACIONESVOLUNTARIAS 13122022/DlgImagFaltante.cpp

Método void CDlgImagFaltante::OnBnClickedGuardarfaltante()

```
....
213.  if( iTipoRechazo == RECHAZO_DEFINITIVO )//Rechazo Normal
```

Insufficiently Protected Credentials

Ruta de consulta:
CPP\Cx\CPP Low Visibility\Insufficiently Protected Credentials Versión:1

Categorías

OWASP Top 10 2013: A6-Sensitive Data Exposure
FISMA 2014: Media Protection
NIST SP 800-53: SC-8 Transmission Confidentiality and Integrity (P1)
OWASP Top 10 2017: A3-Sensitive Data Exposure
ASD STIG 4.10: APSC-DV-001740 - CAT I The application must only store cryptographic representations of passwords.
OWASP Top 10 2021: A4-Insecure Design
MOIS(KISA) Secure Coding 2021: MOIS(KISA) Security Functions
SANS top 25: SANS top 25
CWE top 25: CWE top 25
OWASP ASVS: V02 Authentication

Descripción

Insufficiently Protected Credentials\Ruta 1:

Severidad	Bajas
Estado del resultado	Para verificar
Resultados en línea	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1155844&projectid=45512&pathid=4
Estatus	Nuevo
Detection Date	12/14/2022 10:10:31 AM

	Origen	Destino
Archivo	VALIDADORAPORTACIONESVOLUNTARIAS 13122022/Clases/CConexionOdbc.cpp	VALIDADORAPORTACIONESVOLUNTARIAS 13122022/Clases/CConexionOdbc.cpp

Línea	78	81
Objeto	cPwd	cPwd

Fragmento de código

Nombre del archivo VALIDADORAPORTACIONESVOLUNTARIAS 13122022/Clases/CConexionOdbc.cpp

Método bool CConexionOdbc::conexionPostGreSql(C_ODBC *odbcPar, char *cIp, char *cBd, char *cUsuario)

```
....
78.  char cPwd[40]={0};
....
81.  if( odbcPar->Open("PostgreSQL", cIp, cUsuario, cPwd, cBd )== 1 )
```

Insufficiently Protected Credentials\Ruta 2:

Severidad Bajas

Estado del resultado Para verificar

Resultados en línea https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1155844&projectid=45512&pathid=5

Estatus Nuevo

Detection Date 12/14/2022 10:10:31 AM

	Origen	Destino
Archivo	VALIDADORAPORTACIONESVOLUNTARIAS 13122022/Clases/CConexionOdbc.cpp	VALIDADORAPORTACIONESVOLUNTARIAS 13122022/Clases/CConexionOdbc.cpp
Línea	81	81
Objeto	cPwd	cPwd

Fragmento de código

Nombre del archivo VALIDADORAPORTACIONESVOLUNTARIAS 13122022/Clases/CConexionOdbc.cpp

Método bool CConexionOdbc::conexionPostGreSql(C_ODBC *odbcPar, char *cIp, char *cBd, char *cUsuario)

```
....
81.  if( odbcPar->Open("PostgreSQL", cIp, cUsuario, cPwd, cBd )== 1 )
```

Insufficiently Protected Credentials\Ruta 3:

Severidad Bajas

Estado del resultado Para verificar

Resultados en línea https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1155844&projectid=45512&pathid=6

Estatus Nuevo

Detection Date 12/14/2022 10:10:31 AM

	Origen	Destino

Archivo	VALIDADORAPORTACIONESVOLUNTARIAS 13122022/Clases/CConexionOdbc.cpp	VALIDADORAPORTACIONESVOLUNTARIAS 13122022/Clases/CConexionOdbc.cpp
Línea	79	81
Objeto	cPwd	cPwd

Fragmento de código

Nombre del archivo VALIDADORAPORTACIONESVOLUNTARIAS 13122022/Clases/CConexionOdbc.cpp

Método bool CConexionOdbc::conexionPostGreSql(C_ODBC *odbcPar, char *cIp, char *cBd, char *cUsuario)

```
....
79.  generarPasswordDB( cUsuario, cBd, cPwd);
....
81.  if( odbcPar->Open("PostgreSQL", cIp, cUsuario, cPwd, cBd )== 1 )
```

Insufficiently Protected Credentials\Ruta 4:

Severidad Bajas

Estado del resultado Para verificar

Resultados en línea https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1155844&projectid=45512&pathid=7

Estatus Nuevo

Detection Date 12/14/2022 10:10:31 AM

	Origen	Destino
Archivo	VALIDADORAPORTACIONESVOLUNTARIAS 13122022/Clases/CConexionOdbc.cpp	VALIDADORAPORTACIONESVOLUNTARIAS 13122022/Clases/CConexionOdbc.cpp
Línea	60	63
Objeto	cPsw	cPsw

Fragmento de código

Nombre del archivo VALIDADORAPORTACIONESVOLUNTARIAS 13122022/Clases/CConexionOdbc.cpp

Método bool CConexionOdbc::conexionPostGreSql(C_ODBC *odbcPar, char *cIp, char *cBd, char *cUsuario, char *cPsw)

```
....
60.  bool CConexionOdbc::conexionPostGreSql( C_ODBC *odbcPar, char *cIp,
char *cBd, char *cUsuario, char *cPsw )
....
63.  if( odbcPar->Open("PostgreSQL", cIp, cUsuario, cPsw, cBd )    )
```

Insufficiently Protected Credentials\Ruta 5:

Severidad Bajas

Estado del resultado Para verificar

Resultados en línea https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1155844&projectid=45512&pathid=8

Estatus	Nuevo
Detection Date	12/14/2022 10:10:31 AM

	Origen	Destino
Archivo	VALIDADORAPORTACIONESVOLUNTARIAS 13122022/Clases/CConexionOdbc.cpp	VALIDADORAPORTACIONESVOLUNTARIAS 13122022/Clases/CConexionOdbc.cpp
Línea	63	63
Objeto	cPsw	cPsw

Fragmento de código

Nombre del archivo: VALIDADORAPORTACIONESVOLUNTARIAS 13122022/Clases/CConexionOdbc.cpp

Método: bool CConexionOdbc::conexionPostGreSql(C_ODBC *odbcPar, char *cIp, char *cBd, char *cUsuario, char *cPsw)

```
....
63.  if( odbcPar->Open("PostgreSQL", cIp, cUsuario, cPsw, cBd )    )
```

Insufficiently Protected Credentials\Ruta 6:

Severidad	Bajas
Estado del resultado	Para verificar
Resultados en línea	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1155844&projectid=45512&pathid=9
Estatus	Nuevo
Detection Date	12/14/2022 10:10:31 AM

	Origen	Destino
Archivo	VALIDADORAPORTACIONESVOLUNTARIAS 13122022/Clases/CConexionOdbc.cpp	VALIDADORAPORTACIONESVOLUNTARIAS 13122022/Clases/CConexionOdbc.cpp
Línea	45	46
Objeto	cPwd	cPwd

Fragmento de código

Nombre del archivo: VALIDADORAPORTACIONESVOLUNTARIAS 13122022/Clases/CConexionOdbc.cpp

Método: bool CConexionOdbc::conexionSql(C_ODBC *odbcPar, char *cIp, char *cBd, char *cUsuario)

```
....
45.  char cPwd[40]={0};
46.  generarPasswordDB( cUsuario, cBd, cPwd );
```

Insufficiently Protected Credentials\Ruta 7:

Severidad	Bajas
Estado del resultado	Para verificar
Resultados en línea	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1155844&projectid=

	45512&pathid=10
Estatus	Nuevo
Detection Date	12/14/2022 10:10:31 AM

	Origen	Destino
Archivo	VALIDADORAPORTACIONESVOLUNTARIAS 13122022/Clases/CConexionOdbc.cpp	VALIDADORAPORTACIONESVOLUNTARIAS 13122022/Clases/CConexionOdbc.cpp
Línea	29	29
Objeto	cPsw	cPsw

Fragmento de código

Nombre del archivo	VALIDADORAPORTACIONESVOLUNTARIAS 13122022/Clases/CConexionOdbc.cpp
Método	bool CConexionOdbc::conexionSql(C_ODBC *odbcPar, char *cIp, char *cBd, char *cUsuario, char *cPsw)

```
....
29.  if( odbcPar->OpenWithNoDSN( cIp, cUsuario, cPsw, cBd ) )
```

Insufficiently Protected Credentials\Ruta 8:

Severidad	Bajas
Estado del resultado	Para verificar
Resultados en línea	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1155844&projectid=45512&pathid=11
Estatus	Nuevo
Detection Date	12/14/2022 10:10:31 AM

	Origen	Destino
Archivo	VALIDADORAPORTACIONESVOLUNTARIAS 13122022/Clases/CConexionOdbc.cpp	VALIDADORAPORTACIONESVOLUNTARIAS 13122022/Clases/CConexionOdbc.cpp
Línea	26	29
Objeto	cPsw	cPsw

Fragmento de código

Nombre del archivo	VALIDADORAPORTACIONESVOLUNTARIAS 13122022/Clases/CConexionOdbc.cpp
Método	bool CConexionOdbc::conexionSql(C_ODBC *odbcPar, char *cIp, char *cBd, char *cUsuario, char *cPsw)

```
....
26.  bool CConexionOdbc::conexionSql( C_ODBC *odbcPar, char *cIp, char
*cBd, char *cUsuario, char *cPsw )
....
29.  if( odbcPar->OpenWithNoDSN( cIp, cUsuario, cPsw, cBd ) )
```

Insufficiently Protected Credentials\Ruta 9:

Severidad	Bajas
Estado del resultado	Para verificar
Resultados en línea	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1155844&projectid=45512&pathid=12
Estatus	Nuevo
Detection Date	12/14/2022 10:10:31 AM

	Origen	Destino
Archivo	VALIDADORAPORTACIONESVOLUNTARIAS 13122022/Clases/CConexionOdbc.cpp	VALIDADORAPORTACIONESVOLUNTARIAS 13122022/Clases/CConexionOdbc.cpp
Línea	78	79
Objeto	cPwd	cPwd

Fragmento de código

Nombre del archivo: VALIDADORAPORTACIONESVOLUNTARIAS 13122022/Clases/CConexionOdbc.cpp

Método: bool CConexionOdbc::conexionPostGreSql(C_ODBC *odbcPar, char *cIp, char *cBd, char *cUsuario)

```
....
78.  char cPwd[40]={0};
79.  generarPasswordDB( cUsuario, cBd, cPwd);
```

Insufficiently Protected Credentials\Ruta 10:

Severidad	Bajas
Estado del resultado	Para verificar
Resultados en línea	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1155844&projectid=45512&pathid=13
Estatus	Nuevo
Detection Date	12/14/2022 10:10:31 AM

	Origen	Destino
Archivo	VALIDADORAPORTACIONESVOLUNTARIAS 13122022/Clases/CConexionOdbc.cpp	VALIDADORAPORTACIONESVOLUNTARIAS 13122022/Clases/CConexionOdbc.cpp
Línea	47	47
Objeto	cPwd	cPwd

Fragmento de código

Nombre del archivo: VALIDADORAPORTACIONESVOLUNTARIAS 13122022/Clases/CConexionOdbc.cpp

Método: bool CConexionOdbc::conexionSql(C_ODBC *odbcPar, char *cIp, char *cBd, char *cUsuario)

```
....
47.  if( odbcPar->OpenWithNoDSN( cIp, cUsuario, cPwd, cBd ) )
```

Insufficiently Protected Credentials\Ruta 11:

Severidad	Bajas
Estado del resultado	Para verificar
Resultados en línea	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1155844&projectid=45512&pathid=14
Estatus	Nuevo
Detection Date	12/14/2022 10:10:31 AM

	Origen	Destino
Archivo	VALIDADORAPORTACIONESVOLUNTARIAS 13122022/Clases/CConexionOdbc.cpp	VALIDADORAPORTACIONESVOLUNTARIAS 13122022/Clases/CConexionOdbc.cpp
Línea	45	47
Objeto	cPwd	cPwd

Fragmento de código

Nombre del archivo	VALIDADORAPORTACIONESVOLUNTARIAS 13122022/Clases/CConexionOdbc.cpp
Método	bool CConexionOdbc::conexionSql(C_ODBC *odbcPar, char *cIp, char *cBd, char *cUsuario)

```
....
45.  char cPwd[40]={0};
....
47.  if( odbcPar->OpenWithNoDSN( cIp, cUsuario, cPwd, cBd ) )
```

Insufficiently Protected Credentials\Ruta 12:

Severidad	Bajas
Estado del resultado	Para verificar
Resultados en línea	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1155844&projectid=45512&pathid=15
Estatus	Nuevo
Detection Date	12/14/2022 10:10:31 AM

	Origen	Destino
Archivo	VALIDADORAPORTACIONESVOLUNTARIAS 13122022/Clases/CConexionOdbc.cpp	VALIDADORAPORTACIONESVOLUNTARIAS 13122022/Clases/CConexionOdbc.cpp
Línea	46	47
Objeto	cPwd	cPwd

Fragmento de código

Nombre del archivo	VALIDADORAPORTACIONESVOLUNTARIAS 13122022/Clases/CConexionOdbc.cpp
Método	bool CConexionOdbc::conexionSql(C_ODBC *odbcPar, char *cIp, char *cBd, char *cUsuario)

```
....
46.  generarPasswordDB( cUsuario, cBd, cPwd );
47.  if( odbcPar->OpenWithNoDSN( cIp, cUsuario, cPwd, cBd ) )
```

Use of Sizeof On a Pointer Type

Ruta de consulta:
CPP\Cx\CPP Low Visibility\Use of Sizeof On a Pointer Type Versión:2

Descripción

Use of Sizeof On a Pointer Type\Ruta 1:

Severidad	Bajas
Estado del resultado	Para verificar
Resultados en línea	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1155844&projectid=45512&pathid=16
Estatus	Nuevo
Detection Date	12/14/2022 10:10:32 AM

	Origen	Destino
Archivo	VALIDADORAPORTACIONESVOLUNTARIAS 13122022/DlgAltaOperadores.cpp	VALIDADORAPORTACIONESVOLUNTARIAS 13122022/DlgAltaOperadores.cpp
Línea	468	484
Objeto	cCadena	sizeof

Fragmento de código

Nombre del archivo: VALIDADORAPORTACIONESVOLUNTARIAS 13122022/DlgAltaOperadores.cpp

Método: bool CDlgAltaOperadores::validarControl(char *cCadena)

```
....
468.  bool CDlgAltaOperadores::validarControl( char *cCadena )
....
484.  ZeroMemory(cCadena, sizeof(cCadena));
```

Use of Sizeof On a Pointer Type\Ruta 2:

Severidad	Bajas
Estado del resultado	Para verificar
Resultados en línea	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1155844&projectid=45512&pathid=17
Estatus	Nuevo
Detection Date	12/14/2022 10:10:32 AM

	Origen	Destino
Archivo	VALIDADORAPORTACIONESVOLUNTARIAS 13122022/DlgAltaOperadores.cpp	VALIDADORAPORTACIONESVOLUNTARIAS 13122022/DlgAltaOperadores.cpp
Línea	468	492
Objeto	cCadena	sizeof

Fragmento de código

Nombre del archivo: VALIDADORAPORTACIONESVOLUNTARIAS 13122022/DlgAltaOperadores.cpp

Método bool CDlgAltaOperadores::validarControl(char *cCadena)

```
....
468.  bool CDlgAltaOperadores::validarControl( char *cCadena )
....
492.  ZeroMemory(cCadena, sizeof(cCadena));
```

Use of Sizeof On a Pointer Type\Ruta 3:

Severidad	Bajas
Estado del resultado	Para verificar
Resultados en línea	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1155844&projectid=45512&pathid=18
Estatus	Nuevo
Detection Date	12/14/2022 10:10:32 AM

	Origen	Destino
Archivo	VALIDADORAPORTACIONESVOLUNTARIAS 13122022/DlgAltaOperadores.cpp	VALIDADORAPORTACIONESVOLUNTARIAS 13122022/DlgAltaOperadores.cpp
Línea	468	507
Objeto	cCadena	sizeof

Fragmento de código

Nombre del archivo VALIDADORAPORTACIONESVOLUNTARIAS 13122022/DlgAltaOperadores.cpp

Método bool CDlgAltaOperadores::validarControl(char *cCadena)

```
....
468.  bool CDlgAltaOperadores::validarControl( char *cCadena )
....
507.  ZeroMemory(cCadena, sizeof(cCadena));
```

Use of Sizeof On a Pointer Type\Ruta 4:

Severidad	Bajas
Estado del resultado	Para verificar
Resultados en línea	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1155844&projectid=45512&pathid=19
Estatus	Nuevo
Detection Date	12/14/2022 10:10:32 AM

	Origen	Destino
Archivo	VALIDADORAPORTACIONESVOLUNTARIAS 13122022/DlgIntegridad.cpp	VALIDADORAPORTACIONESVOLUNTARIAS 13122022/DlgIntegridad.cpp
Línea	2410	2432
Objeto	cNomOld	sizeof

Fragmento de código

Nombre del archivo	VALIDADORAPORTACIONESVOLUNTARIAS 13122022/DlgIntegridad.cpp
Método	bool CDlgIntegridad::buscarArchCamNom(char *cNomOld, char *cNomNew)

```
....
2410.  bool CDlgIntegridad::buscarArchCamNom( char *cNomOld, char
*cNomNew )
....
2432.  hr = StringCchPrintf( cNomOld, sizeof(cNomOld), "%s.tif",
sTexto.Mid( 0, 47 ));
```

Use of Sizeof On a Pointer Type\Ruta 5:

Severidad	Bajas
Estado del resultado	Para verificar
Resultados en línea	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1155844&projectid=45512&pathid=20
Estatus	Nuevo
Detection Date	12/14/2022 10:10:32 AM

	Origen	Destino
Archivo	VALIDADORAPORTACIONESVOLUNTARIAS 13122022/DlgIntegridad.cpp	VALIDADORAPORTACIONESVOLUNTARIAS 13122022/DlgIntegridad.cpp
Línea	2410	2436
Objeto	cNomNew	sizeof

Fragmento de código

Nombre del archivo	VALIDADORAPORTACIONESVOLUNTARIAS 13122022/DlgIntegridad.cpp
Método	bool CDlgIntegridad::buscarArchCamNom(char *cNomOld, char *cNomNew)

```
....
2410.  bool CDlgIntegridad::buscarArchCamNom( char *cNomOld, char
*cNomNew )
....
2436.  hr = StringCchPrintf( cNomNew, sizeof(cNomNew), "%s.tif",
sTexto.Mid( 0, 47 ));
```

Use of Inherently Dangerous Function

Weakness ID: 242 *(Weakness Base)* **Status:** Draft

Description

Description Summary

The program calls a function that can never be guaranteed to work safely.

Extended Description

Certain functions behave in dangerous ways regardless of how they are used. Functions in this category were often implemented without taking security concerns into account. The gets() function is unsafe because it does not perform bounds checking on the size of its input. An attacker can easily send arbitrarily-sized input to gets() and overflow the destination buffer. Similarly, the >> operator is unsafe to use when reading into a statically-allocated character array because it does not perform bounds checking on the

size of its input. An attacker can easily send arbitrarily-sized input to the >> operator and overflow the destination buffer.

Time of Introduction

- Implementation

Applicable Platforms

Languages

C

C++

Likelihood of Exploit

High

Demonstrative Examples

Example 1

The excerpt below calls the gets() function in C, which is inherently unsafe.

(Bad Code)

Example Language: **C**

```
char buf[BUFSIZE];
gets(buf);
```

Example 2

The excerpt below calls the gets() function in C, which is inherently unsafe.

(Bad Code)

Example Language: **C**

```
char buf[24];
printf("Please enter your name and press <Enter>\n");
gets(buf);
...
}
```

However, the programmer uses the function gets() which is inherently unsafe because it blindly copies all input from STDIN to the buffer without checking size. This allows the user to provide a string that is larger than the buffer size, resulting in an overflow condition.

Potential Mitigations

Ban the use of dangerous function. Use their safe equivalent.

Use grep or static analysis tools to spot usage of dangerous functions.

Weakness Ordinalities

Ordinality	Description
Primary	*(where the weakness exists independent of other weaknesses)*

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Weakness Class	227	Failure to Fulfill API Contract ('API Abuse')	**Development Concepts (primary)699 Seven Pernicious Kingdoms (primary)700**
ChildOf	Weakness Class	710	Coding Standards Violation	**Research Concepts (primary)1000**
ChildOf	Category	748	CERT C Secure Coding Section 50 - POSIX (POS)	**Weaknesses Addressed by the CERT C Secure Coding Standard**

				(primary)734
CanPrecede	Weakness Base	120	Buffer Copy without Checking Size of Input ('Classic Buffer Overflow')	Research Concepts1000

f Causal Nature

Explicit

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name
7 Pernicious Kingdoms			Dangerous Functions
CERT C Secure Coding	POS33-C		Do not use vfork()

References

Herbert Schildt. "Herb Schildt's C++ Programming Cookbook". Chapter 5. Working with I/O. McGraw-Hill Osborne Media. 2008-04-28.

[REF-11] M. Howard and D. LeBlanc. "Writing Secure Code". Chapter 5, "gets and fgets" Page 163. 2nd Edition. Microsoft. 2002.

Content History

Submissions

Submission Date	Submitter	Organization	Source
	7 Pernicious Kingdoms		Externally Mined

Modifications

Modification Date	Modifier	Organization	Source
2008-07-01	Sean Eidemiller	Cigital	External
	added/updated demonstrative examples		
2008-07-01	Eric Dalci	Cigital	External
	updated Potential Mitigations		
2008-09-08	CWE Content Team	MITRE	Internal
	updated Applicable Platforms, Relationships, Other Notes, Taxonomy Mappings, Type, Weakness Ordinalities		
2008-11-24	CWE Content Team	MITRE	Internal
	updated Relationships, Taxonomy Mappings		
2009-10-29	CWE Content Team	MITRE	Internal
	updated Description, Other Notes, References		
2010-02-16	CWE Content Team	MITRE	Internal
	updated Demonstrative Examples, References, Relationships		
2010-04-05	CWE Content Team	MITRE	Internal
	updated Relationships		

Previous Entry Names

Change Date	Previous Entry Name
2008-01-30	Dangerous Functions
2008-04-11	Use of Inherently Dangerous Functions

BACK TO TOP

Failure to Release Memory Before Removing Last Reference ('Memory Leak')

Weakness ID: 401 *(Weakness Base)* **Status:** Draft

Description

Description Summary

The software does not sufficiently track and release allocated memory after it has been used, which slowly consumes remaining memory.

Extended Description

This is often triggered by improper handling of malformed data or unexpectedly interrupted sessions.

Terminology Notes

"memory leak" has sometimes been used to describe other kinds of issues, e.g. for information leaks in which the contents of memory are inadvertently leaked (CVE-2003-0400 is one such example of this terminology conflict).

Time of Introduction

- Architecture and Design
- Implementation

Applicable Platforms

Languages

C

C++

Modes of Introduction

Memory leaks have two common and sometimes overlapping causes:

- Error conditions and other exceptional circumstances
- Confusion over which part of the program is responsible for freeing the memory

Common Consequences

Scope	Effect
Availability	Most memory leaks result in general software reliability problems, but if an attacker can intentionally trigger a memory leak, the attacker might be able to launch a denial of service attack (by crashing or hanging the program) or take advantage of other unexpected program behavior resulting from a low memory condition.

Likelihood of Exploit

Medium

Demonstrative Examples

Example 1

The following C function leaks a block of allocated memory if the call to read() fails to return the expected number of bytes:

(Bad Code)

Example Language: **C**

```
char* getBlock(int fd) {
char* buf = (char*) malloc(BLOCK_SIZE);
if (!buf) {
return NULL;
}
if (read(fd, buf, BLOCK_SIZE) != BLOCK_SIZE) {

return NULL;
}
```

```
return buf;
}
```

Example 2

Here the problem is that every time a connection is made, more memory is allocated. So if one just opened up more and more connections, eventually the machine would run out of memory.

(Bad Code)

Example Language: **C**

```
bar connection(){
foo = malloc(1024);
return foo;
}
endConnection(bar foo) {

free(foo);
}
int main() {

while(1) //thread 1
//On a connection
foo=connection(); //thread 2
//When the connection ends
endConnection(foo)
}
```

Observed Examples

Reference	Description
CVE-2005-3119	Memory leak because function does not free() an element of a data structure.
CVE-2004-0427	Memory leak when counter variable is not decremented.
CVE-2002-0574	Memory leak when counter variable is not decremented.
CVE-2005-3181	Kernel uses wrong function to release a data structure, preventing data from being properly tracked by other code.
CVE-2004-0222	Memory leak via unknown manipulations as part of protocol test suite.
CVE-2001-0136	Memory leak via a series of the same command.

Potential Mitigations

Pre-design: Use a language or compiler that performs automatic bounds checking.

Phase: Architecture and Design

Use an abstraction library to abstract away risky APIs. Not a complete solution.

Pre-design through Build: The Boehm-Demers-Weiser Garbage Collector or valgrind can be used to detect leaks in code. This is not a complete solution as it is not 100% effective.

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Weakness Class	398	Indicator of Poor Code Quality	**Seven Pernicious Kingdoms (primary)700**
ChildOf	Category	399	Resource Management Errors	**Development Concepts (primary)699**
ChildOf	Category	633	Weaknesses that Affect Memory	**Resource-specific Weaknesses (primary)631**
ChildOf	Category	730	OWASP Top Ten 2004 Category A9 - Denial of Service	**Weaknesses in OWASP Top Ten (2004) (primary)711**
ChildOf	Weakness Base	772	Missing Release of Resource after Effective	**Research Concepts (primary)1000**

			Lifetime	
MemberOf	View	630	Weaknesses Examined by SAMATE	**Weaknesses Examined by SAMATE (primary)630**
CanFollow	Weakness Class	390	Detection of Error Condition Without Action	Research Concepts1000

Relationship Notes

This is often a resultant weakness due to improper handling of malformed data or early termination of sessions.

Affected Resources

- Memory

Functional Areas

- Memory management

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name
PLOVER			Memory leak
7 Pernicious Kingdoms			Memory Leak
CLASP			Failure to deallocate data
OWASP Top Ten 2004	A9	CWE More Specific	Denial of Service

White Box Definitions

A weakness where the code path has:

1. start statement that allocates dynamically allocated memory resource

2. end statement that loses identity of the dynamically allocated memory resource creating situation where dynamically allocated memory resource is never relinquished

Where "loses" is defined through the following scenarios:

1. identity of the dynamic allocated memory resource never obtained

2. the statement assigns another value to the data element that stored the identity of the dynamically allocated memory resource and there are no aliases of that data element

3. identity of the dynamic allocated memory resource obtained but never passed on to function for memory resource release

4. the data element that stored the identity of the dynamically allocated resource has reached the end of its scope at the statement and there are no aliases of that data element

References

J. Whittaker and H. Thompson. "How to Break Software Security". Addison Wesley. 2003.

Content History

Submissions			
Submission Date	**Submitter**	**Organization**	**Source**
	PLOVER		Externally Mined
Modifications			
Modification Date	**Modifier**	**Organization**	**Source**
2008-07-01	Eric Dalci	Cigital	External
	updated Time of Introduction		
2008-08-01		KDM Analytics	External
	added/updated white box definitions		
2008-08-15		Veracode	External
	Suggested OWASP Top Ten 2004 mapping		
2008-09-08	CWE Content Team	MITRE	Internal
	updated Applicable Platforms, Common Consequences, Relationships, Other Notes, References, Relationship Notes, Taxonomy Mappings, Terminology Notes		
2008-10-14	CWE Content Team	MITRE	Internal
	updated Description		
2009-03-10	CWE Content Team	MITRE	Internal
	updated Other Notes		
2009-05-27	CWE Content Team	MITRE	Internal
	updated Name		
2009-07-17	KDM Analytics		External
	Improved the White Box Definition		

2009-07-27	CWE Content Team	MITRE	Internal
	updated White Box Definitions		
2009-10-29	CWE Content Team	MITRE	Internal
	updated Modes of Introduction, Other Notes		
2010-02-16	CWE Content Team	MITRE	Internal
	updated Relationships		

Previous Entry Names

Change Date	Previous Entry Name
2008-04-11	Memory Leak
2009-05-27	Failure to Release Memory Before Removing Last Reference (aka 'Memory Leak')

BACK TO TOP

Use of Uninitialized Variable

Weakness ID: 457 *(Weakness Variant)* **Status:** Draft

Description

Description Summary

The code uses a variable that has not been initialized, leading to unpredictable or unintended results.

Extended Description

In some languages, such as C, an uninitialized variable contains contents of previously-used memory. An attacker can sometimes control or read these contents.

Time of Introduction

- Implementation

Applicable Platforms

Languages

C: *(Sometimes)*

C++: *(Sometimes)*

Perl: *(Often)*

All

Common Consequences

Scope	Effect
Availability Integrity	Initial variables usually contain junk, which can not be trusted for consistency. This can lead to denial of service conditions, or modify control flow in unexpected ways. In some cases, an attacker can "pre-initialize" the variable using previous actions, which might enable code execution. This can cause a race condition if a lock variable check passes when it should not.
Authorization	Strings that are not initialized are especially dangerous, since many functions expect a null at the end -- and only at the end -- of a string.

Likelihood of Exploit

High

Demonstrative Examples

Example 1

The following switch statement is intended to set the values of the variables aN and bN, but in the default case, the programmer has accidentally set the value of aN twice. As a result, bN will have an undefined value.

(Bad Code)

Example Language: **C**

```
switch (ctl) {
case -1:
aN = 0;
bN = 0;
break;
case 0:
aN = i;
bN = -i;
break;
case 1:
aN = i + NEXT_SZ;
bN = i - NEXT_SZ;
break;
default:
```

```
aN = -1;
aN = -1;
break;
}
repaint(aN, bN);
```

Most uninitialized variable issues result in general software reliability problems, but if attackers can intentionally trigger the use of an uninitialized variable, they might be able to launch a denial of service attack by crashing the program. Under the right circumstances, an attacker may be able to control the value of an uninitialized variable by affecting the values on the stack prior to the invocation of the function.

Example 2

Example Languages: **C++ and Java**

```
int foo;
void bar() {
if (foo==0)
/.../
/../
}
```

Observed Examples

Reference	Description
CVE-2008-0081	Uninitialized variable leads to code execution in popular desktop application.
CVE-2007-4682	Crafted input triggers dereference of an uninitialized object pointer.
CVE-2007-3468	Crafted audio file triggers crash when an uninitialized variable is used.
CVE-2007-2728	Uninitialized random seed variable used.

Potential Mitigations

Phase: Implementation

Assign all variables to an initial value.

Phase: Build and Compilation

Most compilers will complain about the use of uninitialized variables if warnings are turned on.

Phase: Requirements

The choice could be made to use a language that is not susceptible to these issues.

Phase: Architecture and Design

Mitigating technologies such as safe string libraries and container abstractions could be introduced.

Other Notes

Before variables are initialized, they generally contain junk data of what was left in the memory that the variable takes up. This data is very rarely useful, and it is generally advised to pre-initialize variables or set them to their first values early. If one forgets -- in the C language -- to initialize, for example a char *, many of the simple string libraries may often return incorrect results as they expect the null termination to be at the end of a string.

Stack variables in C and C++ are not initialized by default. Their initial values are determined by whatever happens to be in their location on the stack at the time the function is invoked. Programs should never use the value of an uninitialized variable. It is not uncommon for programmers to use an uninitialized variable in code that handles errors or other rare and exceptional circumstances. Uninitialized variable warnings can sometimes indicate the presence of a typographic error in the code.

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Weakness Class	398	Indicator of Poor Code Quality	**Seven Pernicious Kingdoms (primary)700**
ChildOf	Weakness Base	456	Missing Initialization	**Development Concepts (primary)699 Research Concepts**

				(primary)1000
MemberOf	View	630	Weaknesses Examined by SAMATE	**Weaknesses Examined by SAMATE (primary)630**

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name
CLASP			Uninitialized variable
7 Pernicious Kingdoms			Uninitialized Variable

White Box Definitions

A weakness where the code path has:

1. start statement that defines variable
2. end statement that accesses the variable
3. the code path does not contain a statement that assigns value to the variable

References

mercy. "Exploiting Uninitialized Data". Jan 2006. < http://www.felinemenace.org/~mercy/papers/UBehavior/UBehavior.zip>.

Microsoft Security Vulnerability Research & Defense. "MS08-014 : The Case of the Uninitialized Stack Variable Vulnerability". 2008-03-11. <http://blogs.technet.com/swi/archive/2008/03/11/the-case-of-the-uninitialized-stack-variable-vulnerability.aspx>.

Content History

Submissions

Submission Date	Submitter	Organization	Source
	CLASP		Externally Mined

Modifications

Modification Date	Modifier	Organization	Source
2008-07-01	Eric Dalci	Cigital	External
	updated Time of Introduction		
2008-08-01		KDM Analytics	External
	added/updated white box definitions		
2008-09-08	CWE Content Team	MITRE	Internal
	updated Applicable Platforms, Common Consequences, Description, Relationships, Observed Example, Other Notes, References, Taxonomy Mappings		
2009-01-12	CWE Content Team	MITRE	Internal
	updated Common Consequences, Demonstrative Examples, Potential Mitigations		
2009-03-10	CWE Content Team	MITRE	Internal
	updated Demonstrative Examples		
2009-05-27	CWE Content Team	MITRE	Internal
	updated Demonstrative Examples		

Previous Entry Names

Change Date	Previous Entry Name
2008-04-11	Uninitialized Variable

BACK TO TOP

Insufficiently Protected Credentials

Weakness ID: 522 *(Weakness Base)* **Status:** Incomplete

Description

Description Summary

This weakness occurs when the application transmits or stores authentication credentials and uses an insecure method that is susceptible to unauthorized interception and/or retrieval.

Time of Introduction

- Architecture and Design
- Implementation

Potential Mitigations

Use an appropriate security mechanism to protect the credentials.

Make appropriate use of cryptography to protect the credentials.

Use industry standards to protect the credentials (e.g. LDAP, keystore, etc.).

Other Notes

Attackers are potentially able to bypass authentication mechanisms, hijack a victim's account, and obtain the role and respective access level of the accounts.

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Category	255	Credentials Management	**Development Concepts (primary)699**
ChildOf	Weakness Class	668	Exposure of Resource to Wrong Sphere	**Research Concepts (primary)1000**
ChildOf	Category	718	OWASP Top Ten 2007 Category A7 - Broken Authentication and Session Management	**Weaknesses in OWASP Top Ten (2007) (primary)629**
ChildOf	Category	724	OWASP Top Ten 2004 Category A3 - Broken Authentication and Session Management	**Weaknesses in OWASP Top Ten (2004) (primary)711**
ParentOf	Weakness Variant	256	Plaintext Storage of a Password	**Development Concepts (primary)699 Research Concepts (primary)1000**
ParentOf	Weakness Base	257	Storing Passwords in a Recoverable Format	**Development Concepts (primary)699 Research Concepts (primary)1000**
ParentOf	Weakness Variant	260	Password in Configuration File	**Development Concepts (primary)699 Research Concepts (primary)1000**
ParentOf	Weakness Variant	523	Unprotected Transport of Credentials	**Development Concepts (primary)699 Research Concepts (primary)1000**
ParentOf	Weakness Variant	549	Missing Password Field Masking	**Research Concepts (primary)1000**
ParentOf	Weakness Variant	555	J2EE Misconfiguration: Plaintext Password in Configuration File	**Research Concepts (primary)1000**
ParentOf	Weakness Variant	620	Unverified Password Change	**Development Concepts**

				(primary)699 Research Concepts (primary)1000

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name
Anonymous Tool Vendor (under NDA)			
OWASP Top Ten 2007	A7	CWE More Specific	Broken Authentication and Session Management
OWASP Top Ten 2004	A3	CWE More Specific	Broken Authentication and Session Management

Related Attack Patterns

CAPEC-ID	Attack Pattern Name	*(CAPEC Version: 1.5)*
50	Password Recovery Exploitation	
102	Session Sidejacking	

Content History

Submissions			
Submission Date	**Submitter**	**Organization**	**Source**
	Anonymous Tool Vendor (under NDA)		Externally Mined

Modifications			
Modification Date	**Modifier**	**Organization**	**Source**
2008-07-01	Eric Dalci	Cigital	External
	updated Potential Mitigations, Time of Introduction		
2008-09-08	CWE Content Team	MITRE	Internal
	updated Relationships, Other Notes, Taxonomy Mappings		
2009-05-27	CWE Content Team	MITRE	Internal
	updated Related Attack Patterns		

BACK TO TOP

Use of sizeof() on a Pointer Type

Weakness ID: 467 *(Weakness Variant)* **Status:** Draft

Description

Description Summary

The code calls sizeof() on a malloced pointer type, which always returns the wordsize/8. This can produce an unexpected result if the programmer intended to determine how much memory has been allocated.

Time of Introduction

- Implementation

Applicable Platforms

Languages

C

C++

Common Consequences

Scope	Effect
Integrity	This error can often cause one to allocate a buffer that is much smaller than what is needed, leading to resultant weaknesses such as buffer overflows.

Likelihood of Exploit

High

Demonstrative Examples

Example 1

Care should be taken to ensure sizeof returns the size of the data structure itself, and not the size of the pointer to the data structure.

In this example, sizeof(foo) returns the size of the pointer.

(Bad Code)

Example Languages: **C and C++**

```
double *foo;
...
foo = (double *)malloc(sizeof(foo));
```

In this example, sizeof(*foo) returns the size of the data structure and not the size of the pointer.

(Good Code)

Example Languages: **C and C++**

```
double *foo;
...
foo = (double *)malloc(sizeof(*foo));
```

Example 2

This example defines a fixed username and password. The AuthenticateUser() function is intended to accept a username and a password from an untrusted user, and check to ensure that it matches the username and password. If the username and password match, AuthenticateUser() is intended to indicate that authentication succeeded.

(Bad Code)

```
/* Ignore CWE-259 (hard-coded password) and CWE-309 (use of password system for authentication) for this example. */

char *username = "admin";
char *pass = "password";

int AuthenticateUser(char *inUser, char *inPass) {
```

```
printf("Sizeof username = %d\n", sizeof(username));
printf("Sizeof pass = %d\n", sizeof(pass));

if (strncmp(username, inUser, sizeof(username))) {
printf("Auth failure of username using sizeof\n");
return(AUTH_FAIL);
}
/* Because of CWE-467, the sizeof returns 4 on many platforms and architectures. */
if (! strncmp(pass, inPass, sizeof(pass))) {
printf("Auth success of password using sizeof\n");
return(AUTH_SUCCESS);
}
else {
printf("Auth fail of password using sizeof\n");
return(AUTH_FAIL);
}
}

int main (int argc, char **argv)
{
int authResult;

if (argc < 3) {
ExitError("Usage: Provide a username and password");
}
authResult = AuthenticateUser(argv[1], argv[2]);
if (authResult != AUTH_SUCCESS) {
ExitError("Authentication failed");
}
else {
DoAuthenticatedTask(argv[1]);
}
}
```

In AuthenticateUser(), because sizeof() is applied to a parameter with an array type, the sizeof() call might return 4 on many modern architectures. As a result, the strncmp() call only checks the first four characters of the input password, resulting in a partial comparison (CWE-187), leading to improper authentication (CWE-287).

Because of the partial comparison, any of these passwords would still cause authentication to succeed for the "admin" user:

(Attack)

```
pass5
passABCDEFGH
passWORD
```

Because only 4 characters are checked, this significantly reduces the search space for an attacker, making brute force attacks more feasible.

The same problem also applies to the username, so values such as "adminXYZ" and "administrator" will succeed for the username.

Potential Mitigations

Phase: Implementation

Use expressions such as "sizeof(*pointer)" instead of "sizeof(pointer)", unless you intend to run sizeof() on a pointer type to gain some platform independence or if you are allocating a variable on the stack.

Other Notes

The use of sizeof() on a pointer can sometimes generate useful information. An obvious case is to find out the wordsize on a platform. More often than not, the appearance of sizeof(pointer) indicates a bug.

Weakness Ordinalities

Ordinality	Description
Primary	*(where the weakness exists independent of other weaknesses)*

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Category	465	Pointer Issues	**Development Concepts (primary)699**
ChildOf	Weakness Class	682	Incorrect Calculation	**Research Concepts (primary)1000**
ChildOf	Category	737	CERT C Secure Coding Section 03 - Expressions (EXP)	**Weaknesses Addressed by the CERT C Secure Coding Standard (primary)734**
ChildOf	Category	740	CERT C Secure Coding Section 06 - Arrays (ARR)	Weaknesses Addressed by the CERT C Secure Coding Standard734
CanPrecede	Weakness Base	131	Incorrect Calculation of Buffer Size	Research Concepts1000

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name
CLASP			Use of sizeof() on a pointer type
CERT C Secure Coding	ARR01-C		Do not apply the sizeof operator to a pointer when taking the size of an array
CERT C Secure Coding	EXP01-C		Do not take the size of a pointer to determine the size of the pointed-to type

White Box Definitions

A weakness where code path has:

1. end statement that passes an identity of a dynamically allocated memory resource to a sizeof operator
2. start statement that allocates the dynamically allocated memory resource

References

Robert Seacord. "EXP01-A. Do not take the sizeof a pointer to determine the size of a type". <https://www.securecoding.cert.org/confluence/display/seccode/EXP01-A.+Do+not+take+the+sizeof+a+pointer+to+determine+the+size+of+a+type>.

Content History

Submissions			
Submission Date	**Submitter**	**Organization**	**Source**
	CLASP		Externally Mined
Modifications			
Modification Date	**Modifier**	**Organization**	**Source**
2008-07-01	Eric Dalci	Cigital	External
	updated Time of Introduction		
2008-08-01		KDM Analytics	External
	added/updated white box definitions		
2008-09-08	CWE Content Team	MITRE	Internal
	updated Applicable Platforms, Common Consequences, Relationships, Other Notes, Taxonomy Mappings, Weakness Ordinalities		
2008-11-24	CWE Content Team	MITRE	Internal
	updated Relationships, Taxonomy Mappings		
2009-03-10	CWE Content Team	MITRE	Internal
	updated Demonstrative Examples		
2009-12-28	CWE Content Team	MITRE	Internal
	updated Demonstrative Examples		
2010-02-16	CWE Content Team	MITRE	Internal
	updated Relationships		

BACK TO TOP

Lenguajes escaneados

Lenguajes	Número hash	Cambiar fecha
CPP	0713743801493748	06/11/2022
JavaScript	7563418033838805	18/09/2022
VbScript	0386000544005133	04/11/2021
PLSQL	4873116881329330	30/01/2022
Common	0244173337933271	18/09/2022