ReporteDoctoSeguroAfirme-CA0068 Scan Report

Project Name	ReporteDoctoSeguroAfirme-CA0068
Scan Start	Wednesday, December 20, 2023 10:49:02 AM
Preset	Coppel Default Test
Scan Time	00h:00m:30s
Lines Of Code Scanned	1363
Files Scanned	16
Report Creation Time	Wednesday, December 20, 2023 10:52:11 AM
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185
Team	Desarrollo_Sist III
Checkmarx Version	9.6.1.1001 HF2
Scan Type	Full
Source Origin	LocalPath
Density	8/100 (Vulnerabilities/LOC)
Visibility	Public
Scan Custom Fields	Criticidad:1

Filter Settings

Severity

Included: High, Medium, Low, Information

Excluded: None

Result State

Included: To Verify, Not Exploitable, Confirmed, Urgent, Proposed Not Exploitable

Excluded: None

Assigned to

Included: All

Categories

Included:

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

ASA Premium	All
Top Tier	All
ASD STIG 5.2	All
PCI DSS v4.0	All

Excluded:

Uncategorized	None
Custom	None
PCI DSS v3.2.1	None
OWASP Top 10 2013	None
FISMA 2014	None
NIST SP 800-53	None
OWASP Top 10 2017	None
OWASP Mobile Top 10 2016	None
OWASP Top 10 API	None
ASD STIG 4.10	None
OWASP Top 10 2010	None
OWASP Top 10 2021	None
MOIS(KISA) Secure Coding 2021	None
SANS top 25	None
CWE top 25	None
OWASP ASVS	None
ASA Mobile Premium	None
ASA Premium	None
Top Tier	None
ASD STIG 5.2	None
PCI DSS v4.0	None

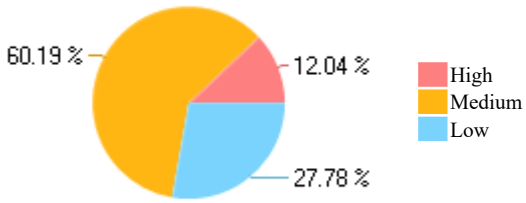
Results Limit

Results limit per query was set to 50

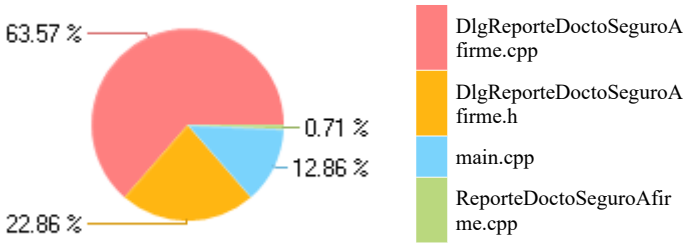
Selected Queries

Selected queries are listed in Result Summary

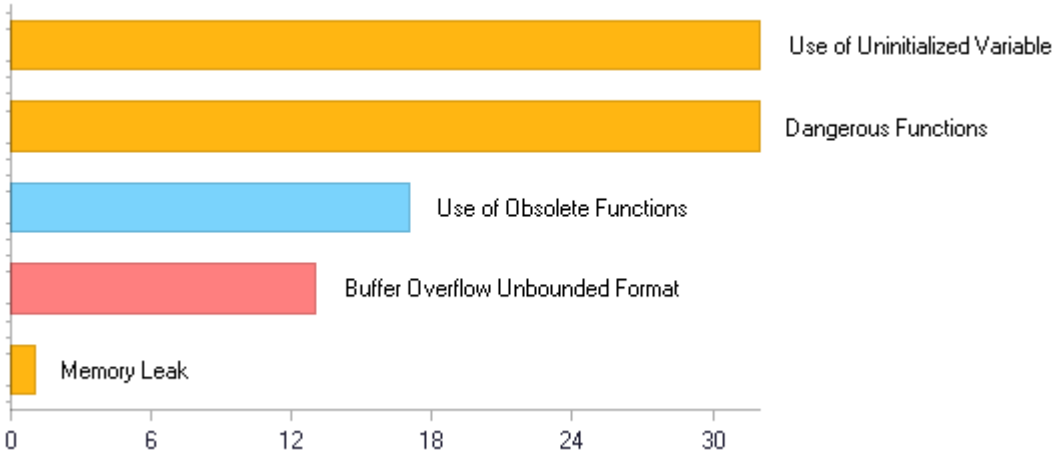
Result Summary



Most Vulnerable Files



Top 5 Vulnerabilities



Scan Summary - OWASP Top 10 2017

Further details and elaboration about vulnerabilities and risks can be found at: 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	0	0
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	49	49
A10-Insufficient Logging & Monitoring	App. Specific	AVERAGE	WIDESPREAD	DIFFICULT	MODERATE	App. Specific	0	0

* Project scan results do not include all relevant queries. Presets and\or Filters should be changed to include all relevant standard queries.

Scan Summary - 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*	94	75
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

* Project scan results do not include all relevant queries. Presets and\or Filters should be changed to include all relevant standard queries.

Scan Summary - OWASP Top 10 2013

Further details and elaboration about vulnerabilities and risks can be found at: 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	0	0
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	49	49
A10-Unvalidated Redirects and Forwards	USERS BROWSERS	AVERAGE	WIDESPREAD	DIFFICULT	MODERATE	AFFECTED DATA AND FUNCTIONS	0	0

* Project scan results do not include all relevant queries. Presets and\or Filters should be changed to include all relevant standard queries.

Scan Summary - 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

* Project scan results do not include all relevant queries. Presets and\or Filters should be changed to include all relevant standard queries.

Scan Summary - 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.	0	0
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

* Project scan results do not include all relevant queries. Presets and\or Filters should be changed to include all relevant standard queries.

Scan Summary - 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)*	33	14
SC-8 Transmission Confidentiality and Integrity (P1)	0	0
SI-10 Information Input Validation (P1)*	0	0
SI-11 Error Handling (P2)*	13	13
SI-15 Information Output Filtering (P0)*	0	0
SI-16 Memory Protection (P1)*	0	0

* Project scan results do not include all relevant queries. Presets and\or Filters should be changed to include all relevant standard queries.

Scan Summary - 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 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.	0	0
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	0	0

	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.		

Scan Summary - 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

Scan Summary - Custom

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

Scan Summary - PCI DSS v4.0

Category	Issues Found	Best Fix Locations
PCI DSS (4.0) - 6.2.4 Vulnerabilities in software development*	26	26

* Project scan results do not include all relevant queries. Presets and\or Filters should be changed to include all relevant standard queries.

Scan Summary - 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	0	0

second for a minimum degree of precision.		
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.	0	0
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	0	0

and diagnostic sessions for organization-defined auditable events.		
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.	0	0
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 product will be configured by following available guidance.	0	0
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 having organization-defined security attribute values with information in transmission.	0	0
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	0	0

all elements of the message must be digitally signed.		
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

Scan Summary - ASD STIG 5.2

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	0	0

second for a minimum degree of precision.		
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.	0	0
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	0	0

and diagnostic sessions for organization-defined auditable events.		
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.*	33	14
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.	13	13
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 product will be configured by following available guidance.	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 having organization-defined security attribute values with information in transmission.	0	0
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	0	0

SAML assertions.		
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

***** Project scan results do not include all relevant queries. Presets and\or Filters should be changed to include all relevant standard queries.

Scan Summary - 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

* Project scan results do not include all relevant queries. Presets and\or Filters should be changed to include all relevant standard queries.

Scan Summary - MOIS(KISA) Secure Coding 2021

Category	Issues Found	Best Fix Locations
MOIS(KISA) API misuse	32	32
MOIS(KISA) Code error*	32	13
MOIS(KISA) Encapsulation	0	0
MOIS(KISA) Error processing*	0	0
MOIS(KISA) Security Functions	0	0
MOIS(KISA) Time and status	0	0
MOIS(KISA) Verification and representation of input data*	13	13

* Project scan results do not include all relevant queries. Presets and\or Filters should be changed to include all relevant standard queries.

Scan Summary - SANS top 25

Category	Issues Found	Best Fix Locations
SANS top 25*	45	26

* Project scan results do not include all relevant queries. Presets and\or Filters should be changed to include all relevant standard queries.

Scan Summary - CWE top 25

Category	Issues Found	Best Fix Locations
CWE top 25*	45	26

* Project scan results do not include all relevant queries. Presets and\or Filters should be changed to include all relevant standard queries.

Scan Summary - Top Tier

Category	Issues Found	Best Fix Locations
Top Tier*	13	13

* Project scan results do not include all relevant queries. Presets and\or Filters should be changed to include all relevant standard queries.

Scan Summary - OWASP ASVS

Category	Issues Found	Best Fix Locations
V01 Architecture, Design and Threat Modeling*	17	17
V02 Authentication	0	0
V03 Session Management	0	0
V04 Access Control	0	0
V05 Validation, Sanitization and Encoding*	13	13
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

* Project scan results do not include all relevant queries. Presets and\or Filters should be changed to include all relevant standard queries.

Scan Summary - ASA Mobile Premium

Category	Issues Found	Best Fix Locations
ASA Mobile Premium	0	0

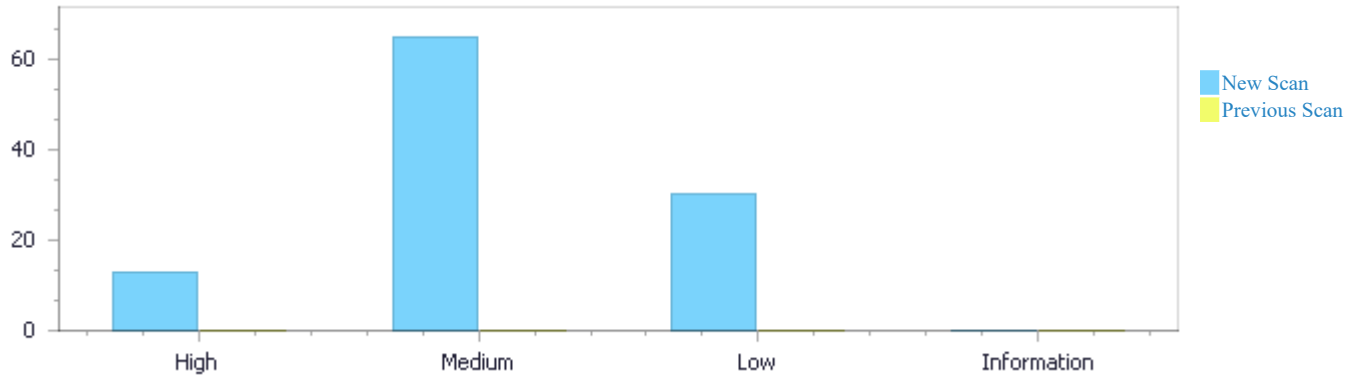
Scan Summary - ASA Premium

Category	Issues Found	Best Fix Locations
ASA Premium*	0	0

* Project scan results do not include all relevant queries. Presets and\or Filters should be changed to include all relevant standard queries.

Results Distribution By Status First scan of the project

	High	Medium	Low	Information	Total
New Issues	13	65	30	0	108
Recurrent Issues	0	0	0	0	0
Total	13	65	30	0	108
Fixed Issues	0	0	0	0	0



Results Distribution By State

	High	Medium	Low	Information	Total
To Verify	13	65	30	0	108
Not Exploitable	0	0	0	0	0
Confirmed	0	0	0	0	0
Urgent	0	0	0	0	0
Proposed Not Exploitable	0	0	0	0	0
Total	13	65	30	0	108

Result Summary

Vulnerability Type	Occurrences	Severity
Buffer Overflow Unbounded Format	13	High
Dangerous Functions	32	Medium
Use of Uninitialized Variable	32	Medium
Memory Leak	1	Medium
Use of Obsolete Functions	17	Low
Unchecked Return Value	13	Low

10 Most Vulnerable Files

High and Medium Vulnerabilities

File Name	Issues Found
DlgReporteDoctoSeguroAfirme.cpp	66
DlgReporteDoctoSeguroAfirme.h	32
main.cpp	11
ReporteDoctoSeguroAfirme.cpp	1

Scan Results Details

Buffer Overflow Unbounded Format

Query Path:
CPP\Cx\CPP Buffer Overflow\Buffer Overflow Unbounded Format Version:3

Categories

OWASP Top 10 2021: A4-Insecure Design
MOIS(KISA) Secure Coding 2021: MOIS(KISA) Verification and representation of input data
SANS top 25: SANS top 25
CWE top 25: CWE top 25
OWASP ASVS: V05 Validation, Sanitization and Encoding
Top Tier: Top Tier
PCI DSS v4.0: PCI DSS (4.0) - 6.2.4 Vulnerabilities in software development

Description

Buffer Overflow Unbounded Format\Path 1:

Severity	High
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=1
Status	New
Detection Date	12/20/2023 10:49:32 AM

The parameter sServer created in DlgReporteDoctoSeguroAfirme.cpp at line 946 is written to a format string in DlgReporteDoctoSeguroAfirme.cpp at line 946 by cIPDestino, without proper bounds checks.

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.cpp	DlgReporteDoctoSeguroAfirme.cpp
Line	963	963
Object	sServer	cIPDestino

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.cpp
Method long CDlgReporteDoctoSeguroAfirme::obtenerFolio(int iTipoFolio,int iIncrementarFolio)

```
....
963.  sprintf(cIPDestino,"%s",sServer);
```

Buffer Overflow Unbounded Format\Path 2:

Severity	High
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=2
Status	New
Detection Date	12/20/2023 10:49:32 AM

The parameter sSqlTxt created in DlgReporteDoctoSeguroAfirme.cpp at line 946 is written to a format string in DlgReporteDoctoSeguroAfirme.cpp at line 946 by cConsulta, without proper bounds checks.

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.cpp	DlgReporteDoctoSeguroAfirme.cpp
Line	966	966

Object	sSqlTxt	cConsulta

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.cpp
Method long CDlgReporteDoctoSeguroAfirme::obtenerFolio(int iTipoFolio,int iIncrementarFolio)

```
....
966.  sprintf(cConsulta,"%s",sSqlTxt);
```

Buffer Overflow Unbounded Format\Path 3:

Severity	High
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=3
Status	New
Detection Date	12/20/2023 10:49:32 AM

The parameter sServer created in DlgReporteDoctoSeguroAfirme.cpp at line 773 is written to a format string in DlgReporteDoctoSeguroAfirme.cpp at line 773 by cIPDestino, without proper bounds checks.

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.cpp	DlgReporteDoctoSeguroAfirme.cpp
Line	793	793
Object	sServer	cIPDestino

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.cpp
Method void CDlgReporteDoctoSeguroAfirme::grabarDocumentosSeguro(long lCliente, long lFolioSeguro)//Checar para que se usaba el pcuentas_cliente->registro_cliente

```
....
793.  sprintf(cIPDestino,"%s",sServer);
```

Buffer Overflow Unbounded Format\Path 4:

Severity	High
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=4
Status	New
Detection Date	12/20/2023 10:49:32 AM

The parameter sSqlTxt created in DlgReporteDoctoSeguroAfirme.cpp at line 773 is written to a format string in DlgReporteDoctoSeguroAfirme.cpp at line 773 by cConsulta, without proper bounds checks.

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.cpp	DlgReporteDoctoSeguroAfirme.cpp
Line	796	796
Object	sSqlTxt	cConsulta

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.cpp

Method	void CDlgReporteDoctoSeguroAfirme::grabarDocumentosSeguro(long lCliente, long lFolioSeguro)//Checar para que se usaba el pcuentas_cliente->registro_cliente

```
....
796.   sprintf(cConsulta,"%s",sSqlTxt);
```

Buffer Overflow Unbounded Format\Path 5:

Severity	High
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=5
Status	New
Detection Date	12/20/2023 10:49:32 AM

The parameter sServer created in DlgReporteDoctoSeguroAfirme.cpp at line 773 is written to a format string in DlgReporteDoctoSeguroAfirme.cpp at line 773 by cIPDestino, without proper bounds checks.

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.cpp	DlgReporteDoctoSeguroAfirme.cpp
Line	837	837
Object	sServer	cIPDestino

Code Snippet	
File Name	DlgReporteDoctoSeguroAfirme.cpp
Method	void CDlgReporteDoctoSeguroAfirme::grabarDocumentosSeguro(long lCliente, long lFolioSeguro)//Checar para que se usaba el pcuentas_cliente->registro_cliente

```
....
837.   sprintf(cIPDestino,"%s",sServer);
```

Buffer Overflow Unbounded Format\Path 6:

Severity	High
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=6
Status	New
Detection Date	12/20/2023 10:49:32 AM

The parameter sSqlTxt created in DlgReporteDoctoSeguroAfirme.cpp at line 773 is written to a format string in DlgReporteDoctoSeguroAfirme.cpp at line 773 by cConsulta, without proper bounds checks.

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.cpp	DlgReporteDoctoSeguroAfirme.cpp
Line	840	840
Object	sSqlTxt	cConsulta

Code Snippet	
File Name	DlgReporteDoctoSeguroAfirme.cpp
Method	void CDlgReporteDoctoSeguroAfirme::grabarDocumentosSeguro(long lCliente, long lFolioSeguro)//Checar para que se usaba el pcuentas_cliente->registro_cliente

```
....
840.  sprintf(cConsulta,"%s",sSqlTxt);
```

Buffer Overflow Unbounded Format\Path 7:

Severity	High
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=7
Status	New
Detection Date	12/20/2023 10:49:32 AM

The parameter sServer created in DlgReporteDoctoSeguroAfirme.cpp at line 773 is written to a format string in DlgReporteDoctoSeguroAfirme.cpp at line 773 by cIPCacarmov, without proper bounds checks.

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.cpp	DlgReporteDoctoSeguroAfirme.cpp
Line	891	891
Object	sServer	cIPCacarmov

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.cpp
Method void CDlgReporteDoctoSeguroAfirme::grabarDocumentosSeguro(long lCliente, long lFolioSeguro)//Checar para que se usaba el pcuentas_cliente->registro_cliente

```
....
891.  sprintf(cIPCacarmov,"%s",sServer);
```

Buffer Overflow Unbounded Format\Path 8:

Severity	High
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=8
Status	New
Detection Date	12/20/2023 10:49:32 AM

The parameter sServer created in DlgReporteDoctoSeguroAfirme.cpp at line 773 is written to a format string in DlgReporteDoctoSeguroAfirme.cpp at line 773 by cIPDestino, without proper bounds checks.

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.cpp	DlgReporteDoctoSeguroAfirme.cpp
Line	902	902
Object	sServer	cIPDestino

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.cpp
Method void CDlgReporteDoctoSeguroAfirme::grabarDocumentosSeguro(long lCliente, long lFolioSeguro)//Checar para que se usaba el pcuentas_cliente->registro_cliente

```
....
902.  sprintf(cIPDestino,"%s",sServer);
```

Buffer Overflow Unbounded Format\Path 9:

Severity	High
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=9
Status	New
Detection Date	12/20/2023 10:49:32 AM

The parameter sSqlTxt created in DlgReporteDoctoSeguroAfirme.cpp at line 773 is written to a format string in DlgReporteDoctoSeguroAfirme.cpp at line 773 by cConsulta, without proper bounds checks.

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.cpp	DlgReporteDoctoSeguroAfirme.cpp
Line	905	905
Object	sSqlTxt	cConsulta

Code Snippet

File Name DlgReporteDoctoSeguroAfirme.cpp

Method void CDlgReporteDoctoSeguroAfirme::grabarDocumentosSeguro(long lCliente, long lFolioSeguro)//Checar para que se usaba el pcuentas_cliente->registro_cliente

```
....
905.  sprintf(cConsulta,"%s",sSqlTxt);
```

Buffer Overflow Unbounded Format\Path 10:

Severity	High
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=10
Status	New
Detection Date	12/20/2023 10:49:32 AM

The parameter sServer created in DlgReporteDoctoSeguroAfirme.cpp at line 918 is written to a format string in DlgReporteDoctoSeguroAfirme.cpp at line 918 by cIPDestino, without proper bounds checks.

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.cpp	DlgReporteDoctoSeguroAfirme.cpp
Line	932	932
Object	sServer	cIPDestino

Code Snippet

File Name DlgReporteDoctoSeguroAfirme.cpp

Method bool CDlgReporteDoctoSeguroAfirme::transaccionGrabarDocumentosSeguro()

```
....
932.  sprintf(cIPDestino,"%s",sServer);
```

Buffer Overflow Unbounded Format\Path 11:

Severity	High
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=11
Status	New

Detection Date	12/20/2023 10:49:32 AM

The parameter sSqlTxt created in DlgReporteDoctoSeguroAfirme.cpp at line 918 is written to a format string in DlgReporteDoctoSeguroAfirme.cpp at line 918 by cConsulta, without proper bounds checks.

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.cpp	DlgReporteDoctoSeguroAfirme.cpp
Line	935	935
Object	sSqlTxt	cConsulta

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.cpp
Method bool CDlgReporteDoctoSeguroAfirme::transaccionGrabarDocumentosSeguro()

```
....
935.  sprintf(cConsulta,"%s",sSqlTxt);
```

Buffer Overflow Unbounded Format\Path 12:

Severity	High
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=12
Status	New
Detection Date	12/20/2023 10:49:32 AM

The parameter sServer created in main.cpp at line 18 is written to a format string in main.cpp at line 18 by cIpTiendaLocal, without proper bounds checks.

	Source	Destination
File	main.cpp	main.cpp
Line	47	47
Object	sServer	cIpTiendaLocal

Code Snippet
File Name main.cpp
Method int CA0068(char *cInput1,char *cInput2)

```
....
47.  sprintf(cIpTiendaLocal, "%s", sServer);
```

Buffer Overflow Unbounded Format\Path 13:

Severity	High
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=13
Status	New
Detection Date	12/20/2023 10:49:32 AM

The parameter sServer created in main.cpp at line 18 is written to a format string in main.cpp at line 18 by cServer, without proper bounds checks.

	Source	Destination
File	main.cpp	main.cpp

Line	48	48
Object	sServer	cServer

Code Snippet
File Name main.cpp
Method int CA0068(char *cInput1,char *cInput2)

```
....
48.  sprintf(cServer, "%s", sServer);
```

Dangerous Functions

Query Path:
CPP\Cx\CPP Medium Threat\Dangerous Functions Version:2

Categories

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

Description
Dangerous Functions\Path 1:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=14
Status	New
Detection Date	12/20/2023 10:49:32 AM

The dangerous function, sprintf, was found in use at line 946 in DlgReporteDoctoSeguroAfirme.cpp file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.cpp	DlgReporteDoctoSeguroAfirme.cpp
Line	963	963
Object	sprintf	sprintf

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.cpp
Method long CDlgReporteDoctoSeguroAfirme::obtenerFolio(int iTipoFolio,int iIncrementarFolio)

```
....
963.  sprintf(cIPDestino,"%s",sServer);
```

Dangerous Functions\Path 2:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=15
Status	New
Detection Date	12/20/2023 10:49:32 AM

The dangerous function, sprintf, was found in use at line 946 in DlgReporteDoctoSeguroAfirme.cpp file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.cpp	DlgReporteDoctoSeguroAfirme.cpp
Line	966	966
Object	sprintf	sprintf

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.cpp
Method long CDlgReporteDoctoSeguroAfirme::obtenerFolio(int iTipoFolio,int iIncrementarFolio)

```
....
966.  sprintf(cConsulta,"%s",sSqlTxt);
```

Dangerous Functions\Path 3:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=16
Status	New
Detection Date	12/20/2023 10:49:32 AM

The dangerous function, memcpy, was found in use at line 473 in DlgReporteDoctoSeguroAfirme.cpp file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.cpp	DlgReporteDoctoSeguroAfirme.cpp
Line	491	491
Object	memcpy	memcpy

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.cpp
Method bool CDlgReporteDoctoSeguroAfirme::validarControl(char *cCadena)

```
....
491.  memcpy(cCadena,cMensajeOut,16);
```

Dangerous Functions\Path 4:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=17
Status	New
Detection Date	12/20/2023 10:49:32 AM

The dangerous function, memcpy, was found in use at line 473 in DlgReporteDoctoSeguroAfirme.cpp file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.cpp	DlgReporteDoctoSeguroAfirme.cpp
Line	500	500

Object	memcpy	memcpy

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.cpp
Method bool CDlgReporteDoctoSeguroAfirme::validarControl(char *cCadena)

```
....
500.  memcpy(cCadena,cMensajeOut,32);
```

Dangerous Functions\Path 5:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=18
Status	New
Detection Date	12/20/2023 10:49:32 AM

The dangerous function, memcpy, was found in use at line 473 in DlgReporteDoctoSeguroAfirme.cpp file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.cpp	DlgReporteDoctoSeguroAfirme.cpp
Line	519	519
Object	memcpy	memcpy

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.cpp
Method bool CDlgReporteDoctoSeguroAfirme::validarControl(char *cCadena)

```
....
519.  memcpy(cCadena,cMensajeOut,41);
```

Dangerous Functions\Path 6:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=19
Status	New
Detection Date	12/20/2023 10:49:32 AM

The dangerous function, sprintf, was found in use at line 773 in DlgReporteDoctoSeguroAfirme.cpp file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.cpp	DlgReporteDoctoSeguroAfirme.cpp
Line	793	793
Object	sprintf	sprintf

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.cpp
Method void CDlgReporteDoctoSeguroAfirme::grabarDocumentosSeguro(long lCliente, long lFolioSeguro)//Checar para que se usaba el pcuentas_cliente->registro_cliente

```
....
793.  sprintf(cIPDestino,"%s",sServer);
```

Dangerous Functions\Path 7:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=20
Status	New
Detection Date	12/20/2023 10:49:32 AM

The dangerous function, sprintf, was found in use at line 773 in DlgReporteDoctoSeguroAfirme.cpp file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.cpp	DlgReporteDoctoSeguroAfirme.cpp
Line	796	796
Object	sprintf	sprintf

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.cpp
Method void CDlgReporteDoctoSeguroAfirme::grabarDocumentosSeguro(long lCliente, long lFolioSeguro)//Checar para que se usaba el pcuentas_cliente->registro_cliente

```
....
796.  sprintf(cConsulta,"%s",sSqlTxt);
```

Dangerous Functions\Path 8:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=21
Status	New
Detection Date	12/20/2023 10:49:32 AM

The dangerous function, sprintf, was found in use at line 773 in DlgReporteDoctoSeguroAfirme.cpp file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.cpp	DlgReporteDoctoSeguroAfirme.cpp
Line	837	837
Object	sprintf	sprintf

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.cpp
Method void CDlgReporteDoctoSeguroAfirme::grabarDocumentosSeguro(long lCliente, long lFolioSeguro)//Checar para que se usaba el pcuentas_cliente->registro_cliente

```
....
837.  sprintf(cIPDestino,"%s",sServer);
```

Dangerous Functions\Path 9:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=22
Status	New
Detection Date	12/20/2023 10:49:32 AM

The dangerous function, sprintf, was found in use at line 773 in DlgReporteDoctoSeguroAfirme.cpp file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.cpp	DlgReporteDoctoSeguroAfirme.cpp
Line	840	840
Object	sprintf	sprintf

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.cpp
Method void CDlgReporteDoctoSeguroAfirme::grabarDocumentosSeguro(long lCliente, long lFolioSeguro)//Checar para que se usaba el pcuentas_cliente->registro_cliente

```
....
840.  sprintf(cConsulta,"%s",sSqlTxt);
```

Dangerous Functions\Path 10:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=23
Status	New
Detection Date	12/20/2023 10:49:32 AM

The dangerous function, memcpy, was found in use at line 773 in DlgReporteDoctoSeguroAfirme.cpp file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.cpp	DlgReporteDoctoSeguroAfirme.cpp
Line	855	855
Object	memcpy	memcpy

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.cpp
Method void CDlgReporteDoctoSeguroAfirme::grabarDocumentosSeguro(long lCliente, long lFolioSeguro)//Checar para que se usaba el pcuentas_cliente->registro_cliente

```
....
855.  memcpy(&grabarDocumentoSeguro2SQL.clavelocal,"1",1);
```

Dangerous Functions\Path 11:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=24

Status	New
Detection Date	12/20/2023 10:49:32 AM

The dangerous function, memcpy, was found in use at line 773 in DlgReporteDoctoSeguroAfirme.cpp file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.cpp	DlgReporteDoctoSeguroAfirme.cpp
Line	860	860
Object	memcpy	memcpy

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.cpp
Method void CDlgReporteDoctoSeguroAfirme::grabarDocumentosSeguro(long lCliente, long lFolioSeguro)//Checar para que se usaba el pcuentas_cliente->registro_cliente

```
....
860.  memcpy(&grabarDocumentoSeguro2SQL.clavelocal,"2",1);
```

Dangerous Functions\Path 12:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=25
Status	New
Detection Date	12/20/2023 10:49:32 AM

The dangerous function, memcpy, was found in use at line 773 in DlgReporteDoctoSeguroAfirme.cpp file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.cpp	DlgReporteDoctoSeguroAfirme.cpp
Line	864	864
Object	memcpy	memcpy

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.cpp
Method void CDlgReporteDoctoSeguroAfirme::grabarDocumentosSeguro(long lCliente, long lFolioSeguro)//Checar para que se usaba el pcuentas_cliente->registro_cliente

```
....
864.  memcpy(&grabarDocumentoSeguro2SQL.clave,"G",1);
```

Dangerous Functions\Path 13:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=26
Status	New
Detection Date	12/20/2023 10:49:32 AM

The dangerous function, memcpy, was found in use at line 773 in DlgReporteDoctoSeguroAfirme.cpp file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.cpp	DlgReporteDoctoSeguroAfirme.cpp
Line	867	867
Object	memcpy	memcpy

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.cpp
Method void CDlgReporteDoctoSeguroAfirme::grabarDocumentosSeguro(long lCliente, long lFolioSeguro)//Checar para que se usaba el pcuentas_cliente->registro_cliente

```
....
867.  memcpy(&grabarDocumentoSeguro2SQL.tipomovimiento,"3",1);
```

Dangerous Functions\Path 14:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=27
Status	New
Detection Date	12/20/2023 10:49:32 AM

The dangerous function, memcpy, was found in use at line 773 in DlgReporteDoctoSeguroAfirme.cpp file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.cpp	DlgReporteDoctoSeguroAfirme.cpp
Line	874	874
Object	memcpy	memcpy

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.cpp
Method void CDlgReporteDoctoSeguroAfirme::grabarDocumentosSeguro(long lCliente, long lFolioSeguro)//Checar para que se usaba el pcuentas_cliente->registro_cliente

```
....
874.  memcpy(&grabarDocumentoSeguro2SQL.movtoseguro,"
",1);//tienda_compra
```

Dangerous Functions\Path 15:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=28
Status	New
Detection Date	12/20/2023 10:49:32 AM

The dangerous function, sprintf, was found in use at line 773 in DlgReporteDoctoSeguroAfirme.cpp file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.cpp	DlgReporteDoctoSeguroAfirme.cpp

Line	891	891
Object	sprintf	sprintf

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.cpp
Method void CDlgReporteDoctoSeguroAfirme::grabarDocumentosSeguro(long lCliente, long lFolioSeguro)//Checar para que se usaba el pcuentas_cliente->registro_cliente

```
....
891.  sprintf(cIPCacarmov,"%s",sServer);
```

Dangerous Functions\Path 16:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=29
Status	New
Detection Date	12/20/2023 10:49:32 AM

The dangerous function, memcpy, was found in use at line 773 in DlgReporteDoctoSeguroAfirme.cpp file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.cpp	DlgReporteDoctoSeguroAfirme.cpp
Line	894	894
Object	memcpy	memcpy

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.cpp
Method void CDlgReporteDoctoSeguroAfirme::grabarDocumentosSeguro(long lCliente, long lFolioSeguro)//Checar para que se usaba el pcuentas_cliente->registro_cliente

```
....
894.  memcpy(&grabarDocumentoSeguro2SQL.ipcarteracliente ,sServer,15);
```

Dangerous Functions\Path 17:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=30
Status	New
Detection Date	12/20/2023 10:49:32 AM

The dangerous function, sprintf, was found in use at line 773 in DlgReporteDoctoSeguroAfirme.cpp file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.cpp	DlgReporteDoctoSeguroAfirme.cpp
Line	902	902
Object	sprintf	sprintf

Code Snippet

File Name	DlgReporteDoctoSeguroAfirme.cpp
Method	void CDlgReporteDoctoSeguroAfirme::grabarDocumentosSeguro(long lCliente, long lFolioSeguro)//Checar para que se usaba el pcuentas_cliente->registro_cliente

```
....
902.  sprintf(cIPDestino,"%s",sServer);
```

Dangerous Functions\Path 18:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=31
Status	New
Detection Date	12/20/2023 10:49:32 AM

The dangerous function, sprintf, was found in use at line 773 in DlgReporteDoctoSeguroAfirme.cpp file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.cpp	DlgReporteDoctoSeguroAfirme.cpp
Line	905	905
Object	sprintf	sprintf

Code Snippet	
File Name	DlgReporteDoctoSeguroAfirme.cpp
Method	void CDlgReporteDoctoSeguroAfirme::grabarDocumentosSeguro(long lCliente, long lFolioSeguro)//Checar para que se usaba el pcuentas_cliente->registro_cliente

```
....
905.  sprintf(cConsulta,"%s",sSqlTxt);
```

Dangerous Functions\Path 19:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=32
Status	New
Detection Date	12/20/2023 10:49:32 AM

The dangerous function, sprintf, was found in use at line 918 in DlgReporteDoctoSeguroAfirme.cpp file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.cpp	DlgReporteDoctoSeguroAfirme.cpp
Line	932	932
Object	sprintf	sprintf

Code Snippet	
File Name	DlgReporteDoctoSeguroAfirme.cpp
Method	bool CDlgReporteDoctoSeguroAfirme::transaccionGrabarDocumentosSeguro()

```
....
932.  sprintf(cIPDestino,"%s",sServer);
```

Dangerous Functions\Path 20:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=33
Status	New
Detection Date	12/20/2023 10:49:32 AM

The dangerous function, sprintf, was found in use at line 918 in DlgReporteDoctoSeguroAfirme.cpp file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.cpp	DlgReporteDoctoSeguroAfirme.cpp
Line	935	935
Object	sprintf	sprintf

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.cpp
Method bool CDlgReporteDoctoSeguroAfirme::transaccionGrabarDocumentosSeguro()

```
....
935.  sprintf(cConsulta,"%s",sSqlTxt);
```

Dangerous Functions\Path 21:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=34
Status	New
Detection Date	12/20/2023 10:49:32 AM

The dangerous function, memcpy, was found in use at line 18 in main.cpp file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	main.cpp	main.cpp
Line	35	35
Object	memcpy	memcpy

Code Snippet
File Name main.cpp
Method int CA0068(char *cInput1,char *cInput2)

```
....
35.  memcpy( &cParametros, cInput1, sizeof( SParametros ) );
```

Dangerous Functions\Path 22:

Severity	Medium

Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=35
Status	New
Detection Date	12/20/2023 10:49:32 AM

The dangerous function, sprintf, was found in use at line 18 in main.cpp file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	main.cpp	main.cpp
Line	47	47
Object	sprintf	sprintf

Code Snippet
File Name main.cpp
Method int CA0068(char *cInput1,char *cInput2)

```
....
47.  sprintf(cIpTiendaLocal, "%s", sServer);
```

Dangerous Functions\Path 23:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=36
Status	New
Detection Date	12/20/2023 10:49:32 AM

The dangerous function, sprintf, was found in use at line 18 in main.cpp file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	main.cpp	main.cpp
Line	48	48
Object	sprintf	sprintf

Code Snippet
File Name main.cpp
Method int CA0068(char *cInput1,char *cInput2)

```
....
48.  sprintf(cServer, "%s", sServer);
```

Dangerous Functions\Path 24:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=37
Status	New
Detection Date	12/20/2023 10:49:32 AM

The dangerous function, atoi, was found in use at line 87 in DlgReporteDoctoSeguroAfirme.cpp file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.cpp	DlgReporteDoctoSeguroAfirme.cpp
Line	178	178
Object	atoi	atoi

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.cpp
Method BEGIN_MESSAGE_MAP(CDlgReporteDoctoSeguroAfirme, CDialogoML)

```
....
178.  if ( !consultarCiudad( atoi(sTienda), iCiudad, cNombreCiudad,
&odbcTiendaNumero ) )
```

Dangerous Functions\Path 25:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=38
Status	New
Detection Date	12/20/2023 10:49:32 AM

The dangerous function, atol, was found in use at line 473 in DlgReporteDoctoSeguroAfirme.cpp file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.cpp	DlgReporteDoctoSeguroAfirme.cpp
Line	485	485
Object	atol	atol

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.cpp
Method bool CDlgReporteDoctoSeguroAfirme::validarControl(char *cCadena)

```
....
485.  lFolioSeguro = atol(sFolioSeguro);
```

Dangerous Functions\Path 26:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=39
Status	New
Detection Date	12/20/2023 10:49:32 AM

The dangerous function, atoi, was found in use at line 773 in DlgReporteDoctoSeguroAfirme.cpp file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.cpp	DlgReporteDoctoSeguroAfirme.cpp

Line	877	877
Object	atoi	atoi

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.cpp
Method void CDlgReporteDoctoSeguroAfirme::grabarDocumentosSeguro(long lCliente, long lFolioSeguro)//Checar para que se usaba el pcuentas_cliente->registro_cliente

```
....
877.  grabarDocumentoSeguro2SQL.tienda = (short int) atoi(sTienda);
```

Dangerous Functions\Path 27:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=40
Status	New
Detection Date	12/20/2023 10:49:32 AM

The dangerous function, atoi, was found in use at line 18 in main.cpp file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	main.cpp	main.cpp
Line	52	52
Object	atoi	atoi

Code Snippet
File Name main.cpp
Method int CA0068(char *cInput1,char *cInput2)

```
....
52.  iTienda = atoi(sTienda);
```

Dangerous Functions\Path 28:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=41
Status	New
Detection Date	12/20/2023 10:49:32 AM

The dangerous function, atoi, was found in use at line 18 in main.cpp file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	main.cpp	main.cpp
Line	59	59
Object	atoi	atoi

Code Snippet
File Name main.cpp

Method	int CA0068(char *cInput1,char *cInput2)

```
....
59.  dlg.iCaja = atoi(sCaja);
```

Dangerous Functions\Path 29:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=42
Status	New
Detection Date	12/20/2023 10:49:32 AM

The dangerous function, atoi, was found in use at line 18 in main.cpp file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	main.cpp	main.cpp
Line	63	63
Object	atoi	atoi

Code Snippet	
File Name	main.cpp
Method	int CA0068(char *cInput1,char *cInput2)

```
....
63.  dlg.iMuestraMsg = atoi(sMuestraMsg);
```

Dangerous Functions\Path 30:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=43
Status	New
Detection Date	12/20/2023 10:49:32 AM

The dangerous function, atol, was found in use at line 18 in main.cpp file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	main.cpp	main.cpp
Line	66	66
Object	atol	atol

Code Snippet	
File Name	main.cpp
Method	int CA0068(char *cInput1,char *cInput2)

```
....
66.  dlg.lCliente = atol(sCliente);
```

Dangerous Functions\Path 31:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=44
Status	New
Detection Date	12/20/2023 10:49:32 AM

The dangerous function, atol, was found in use at line 18 in main.cpp file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	main.cpp	main.cpp
Line	69	69
Object	atol	atol

Code Snippet
File Name main.cpp
Method int CA0068(char *cInput1,char *cInput2)

```
....
69.  dlg.lFactura1 = atol(sFactura1);
```

Dangerous Functions\Path 32:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=45
Status	New
Detection Date	12/20/2023 10:49:32 AM

The dangerous function, atol, was found in use at line 18 in main.cpp file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	main.cpp	main.cpp
Line	72	72
Object	atol	atol

Code Snippet
File Name main.cpp
Method int CA0068(char *cInput1,char *cInput2)

```
....
72.  dlg.lFactura2 = atol(sFactura2);
```

Use of Uninitialized Variable

Query Path:
CPP\Cx\CPP Medium Threat\Use of Uninitialized Variable Version:0

Categories

NIST SP 800-53: SC-5 Denial of Service Protection (P1)
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
ASD STIG 5.2: APSC-DV-002400 - CAT II The application must restrict the ability to launch Denial of Service (DoS) attacks against itself or other information systems.

Description

Use of Uninitialized Variable\Path 1:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=47
Status	New
Detection Date	12/20/2023 10:49:32 AM

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.h	DlgReporteDoctoSeguroAfirme.cpp
Line	50	955
Object	iCaja	iCaja

Code Snippet

File Name DlgReporteDoctoSeguroAfirme.h

Method int iCaja,iMuestraMsg, iDiaActual,iMesActual,iAnioActual,iCiudad,

```
....
50.  int iCaja,iMuestraMsg, iDiaActual,iMesActual,iAnioActual,iCiudad,
```

File Name DlgReporteDoctoSeguroAfirme.cpp

Method long CDlgReporteDoctoSeguroAfirme::obtenerFolio(int iTipoFolio,int iIncrementarFolio)

```
....
955.  sSqlTxt.Format("SELECT gnincrementarfolio('C', '%d', '%d', '%d' )
", iCaja, iTipoFolio, iIncrementarFolio );
```

Use of Uninitialized Variable\Path 2:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=48
Status	New
Detection Date	12/20/2023 10:49:32 AM

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.h	DlgReporteDoctoSeguroAfirme.cpp
Line	50	970
Object	iCaja	iCaja

Code Snippet

File Name DlgReporteDoctoSeguroAfirme.h

Method int iCaja,iMuestraMsg, iDiaActual,iMesActual,iAnioActual,iCiudad,

```
....
50.  int iCaja,iMuestraMsg, iDiaActual,iMesActual,iAnioActual,iCiudad,
```

File Name	DlgReporteDoctoSeguroAfirme.cpp
Method	long CDlgReporteDoctoSeguroAfirme::obtenerFolio(int iTipoFolio,int iIncrementarFolio)

```
....
970.  grabarMensajeError( "M", iCaja, cIPDestino,
"ReporteDoctoSeguroAfirme", "CDlgReporteDoctoSeguroAfirme",
"obtenerFolio", cConsulta,lEmpleado,"Error
#3133",folioCoppelSQL.odbc,iMuestraMsg);
```

Use of Uninitialized Variable\Path 3:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=49
Status	New
Detection Date	12/20/2023 10:49:32 AM

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.h	DlgReporteDoctoSeguroAfirme.cpp
Line	50	800
Object	iCaja	iCaja

Code Snippet

File Name	DlgReporteDoctoSeguroAfirme.h
Method	int iCaja,iMuestraMsg, iDiaActual,iMesActual,iAnioActual,iCiudad,

```
....
50.  int iCaja,iMuestraMsg, iDiaActual,iMesActual,iAnioActual,iCiudad,
```

File Name	DlgReporteDoctoSeguroAfirme.cpp
Method	void CDlgReporteDoctoSeguroAfirme::grabarDocumentosSeguro(long lCliente, long lFolioSeguro)//Checar para que se usaba el pcuentas_cliente->registro_cliente

```
....
800.  grabarMensajeError( "C", iCaja, cIPDestino,
"ReporteDoctoSeguroAfirme", "CDlgReporteDoctoSeguroAfirme",
"grabarDocumentoSeguro", cConsulta,lEmpleado,"Error
#3128",grabarDocumentoSeguroSQL.odbc,iMuestraMsg);
```

Use of Uninitialized Variable\Path 4:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=50
Status	New

Detection Date	12/20/2023 10:49:32 AM

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.h	DlgReporteDoctoSeguroAfirme.cpp
Line	50	844
Object	iCaja	iCaja

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.h
Method int iCaja,iMuestraMsg, iDiaActual,iMesActual,iAnioActual,iCiudad,

```
....
50.  int iCaja,iMuestraMsg, iDiaActual,iMesActual,iAnioActual,iCiudad,
```

File Name DlgReporteDoctoSeguroAfirme.cpp

Method void CDlgReporteDoctoSeguroAfirme::grabarDocumentosSeguro(long lCliente, long lFolioSeguro)//Checar para que se usaba el pcuentas_cliente->registro_cliente

```
....
844.  grabarMensajeError( "M", iCaja, cIPDestino,
"EntradaDevolucionCobranza", "CDlgEntradaDevolucionCobranza", "altaUdi",
cConsulta,lEmpleado,"Error
#3130",grabarDocumentoSeguro2SQL.odbc,iMuestraMsg);
```

Use of Uninitialized Variable\Path 5:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=51
Status	New
Detection Date	12/20/2023 10:49:32 AM

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.h	DlgReporteDoctoSeguroAfirme.cpp
Line	50	880
Object	iCaja	iCaja

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.h
Method int iCaja,iMuestraMsg, iDiaActual,iMesActual,iAnioActual,iCiudad,

```
....
50.  int iCaja,iMuestraMsg, iDiaActual,iMesActual,iAnioActual,iCiudad,
```

File Name DlgReporteDoctoSeguroAfirme.cpp

Method void CDlgReporteDoctoSeguroAfirme::grabarDocumentosSeguro(long lCliente, long lFolioSeguro)//Checar para que se usaba el pcuentas_cliente->registro_cliente

```
....
880.  grabarDocumentoSeguro2SQL.caja = short int(iCaja);
```

Use of Uninitialized Variable\Path 6:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=52
Status	New
Detection Date	12/20/2023 10:49:32 AM

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.h	DlgReporteDoctoSeguroAfirme.cpp
Line	50	909
Object	iCaja	iCaja

Code Snippet

File Name DlgReporteDoctoSeguroAfirme.h

Method int iCaja,iMuestraMsg, iDiaActual,iMesActual,iAnioActual,iCiudad,

```
....
50.  int iCaja,iMuestraMsg, iDiaActual,iMesActual,iAnioActual,iCiudad,
```

File Name DlgReporteDoctoSeguroAfirme.cpp

Method void CDlgReporteDoctoSeguroAfirme::grabarDocumentosSeguro(long lCliente, long lFolioSeguro)//Checar para que se usaba el pcuentas_cliente->registro_cliente

```
....
909.  grabarMensajeError( "C", iCaja, cIPDestino,
"ReporteDoctoSeguroAfirme", "CDlgReporteDoctoSeguroAfirme",
"grabarDocumentoSeguro", cConsulta,lEmpleado,"Error
#3131",grabarDocumentoSeguro2SQL.odbc,iMuestraMsg);
```

Use of Uninitialized Variable\Path 7:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=53
Status	New
Detection Date	12/20/2023 10:49:32 AM

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.h	DlgReporteDoctoSeguroAfirme.cpp
Line	50	924
Object	iCaja	iCaja

Code Snippet

File Name DlgReporteDoctoSeguroAfirme.h

Method	int iCaja,iMuestraMsg, iDiaActual,iMesActual,iAnioActual,iCiudad,

```
....
50.  int iCaja,iMuestraMsg, iDiaActual,iMesActual,iAnioActual,iCiudad,
```

File Name	DlgReporteDoctoSeguroAfirme.cpp
Method	bool CDlgReporteDoctoSeguroAfirme::transaccionGrabarDocumentosSeguro()

```
....
924.  sSqlTxt.Format("SELECT cagrabardocumentosseguro('%d')",iCaja);
```

Use of Uninitialized Variable\Path 8:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=54
Status	New
Detection Date	12/20/2023 10:49:32 AM

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.h	DlgReporteDoctoSeguroAfirme.cpp
Line	50	939
Object	iCaja	iCaja

Code Snippet	
File Name	DlgReporteDoctoSeguroAfirme.h
Method	int iCaja,iMuestraMsg, iDiaActual,iMesActual,iAnioActual,iCiudad,

```
....
50.  int iCaja,iMuestraMsg, iDiaActual,iMesActual,iAnioActual,iCiudad,
```

File Name	DlgReporteDoctoSeguroAfirme.cpp
Method	bool CDlgReporteDoctoSeguroAfirme::transaccionGrabarDocumentosSeguro()

```
....
939.  grabarMensajeError( "C", iCaja, cIPDestino,
"ReporteDoctoSeguroAfirme", "CDlgReporteDoctoSeguroAfirme",
"transaccionGrabarDocumentosSeguro", cConsulta,lEmpleado,"ERROR
#3132",grabarDocumentoSeguro2SQL.odbc,iMuestraMsg);
```

Use of Uninitialized Variable\Path 9:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=55
Status	New
Detection Date	12/20/2023 10:49:32 AM

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.h	DlgReporteDoctoSeguroAfirme.cpp
Line	49	970
Object	lEmpleado	lEmpleado

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.h
Method long lFacturaG,lFactura1,lFactura2,lCliente,lFolioSeguro,lEmpleado;

```
....
49.  long lFacturaG,lFactura1,lFactura2,lCliente,lFolioSeguro,lEmpleado;
```

File Name DlgReporteDoctoSeguroAfirme.cpp

Method long CDlgReporteDoctoSeguroAfirme::obtenerFolio(int iTipoFolio,int iIncrementarFolio)

```
....
970.  grabarMensajeError( "M", iCaja, cIPDestino,
"ReporteDoctoSeguroAfirme", "CDlgReporteDoctoSeguroAfirme",
"obtenerFolio", cConsulta,lEmpleado,"Error
#3133",folioCoppelSQL.odbc,iMuestraMsg);
```

Use of Uninitialized Variable\Path 10:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=56
Status	New
Detection Date	12/20/2023 10:49:32 AM

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.h	DlgReporteDoctoSeguroAfirme.cpp
Line	49	800
Object	lEmpleado	lEmpleado

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.h
Method long lFacturaG,lFactura1,lFactura2,lCliente,lFolioSeguro,lEmpleado;

```
....
49.  long lFacturaG,lFactura1,lFactura2,lCliente,lFolioSeguro,lEmpleado;
```

File Name DlgReporteDoctoSeguroAfirme.cpp

Method void CDlgReporteDoctoSeguroAfirme::grabarDocumentosSeguro(long lCliente, long lFolioSeguro)//Checar para que se usaba el pcuentas_cliente->registro_cliente

```
....
800.  grabarMensajeError( "C", iCaja, cIPDestino,
"ReporteDoctoSeguroAfirme", "CDlgReporteDoctoSeguroAfirme",
"grabarDocumentoSeguro", cConsulta,lEmpleado,"Error
#3128",grabarDocumentoSeguroSQL.odbc,iMuestraMsg);
```

Use of Uninitialized Variable\Path 11:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=57
Status	New
Detection Date	12/20/2023 10:49:32 AM

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.h	DlgReporteDoctoSeguroAfirme.cpp
Line	49	844
Object	lEmpleado	lEmpleado

Code Snippet

File Name DlgReporteDoctoSeguroAfirme.h

Method long lFacturaG,lFactura1,lFactura2,lCliente,lFolioSeguro,lEmpleado;

```
....
49.  long lFacturaG,lFactura1,lFactura2,lCliente,lFolioSeguro,lEmpleado;
```

File Name DlgReporteDoctoSeguroAfirme.cpp

Method void CDlgReporteDoctoSeguroAfirme::grabarDocumentosSeguro(long lCliente, long lFolioSeguro)//Checar para que se usaba el pcuentas_cliente->registro_cliente

```
....
844.  grabarMensajeError( "M", iCaja, cIPDestino,
"EntradaDevolucionCobranza", "CDlgEntradaDevolucionCobranza", "altaUdi",
cConsulta,lEmpleado,"Error
#3130",grabarDocumentoSeguro2SQL.odbc,iMuestraMsg);
```

Use of Uninitialized Variable\Path 12:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=58
Status	New
Detection Date	12/20/2023 10:49:32 AM

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.h	DlgReporteDoctoSeguroAfirme.cpp
Line	49	909
Object	lEmpleado	lEmpleado

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.h
Method long lFacturaG,lFactura1,lFactura2,lCliente,lFolioSeguro,lEmpleado;

```
....
49.  long lFacturaG,lFactura1,lFactura2,lCliente,lFolioSeguro,lEmpleado;
```

File Name DlgReporteDoctoSeguroAfirme.cpp

Method void CDlgReporteDoctoSeguroAfirme::grabarDocumentosSeguro(long lCliente, long lFolioSeguro)//Checar para que se usaba el pcuentas_cliente->registro_cliente

```
....
909.  grabarMensajeError( "C", iCaja, cIPDestino,
"ReporteDoctoSeguroAfirme", "CDlgReporteDoctoSeguroAfirme",
"grabarDocumentoSeguro", cConsulta,lEmpleado,"Error
#3131",grabarDocumentoSeguro2SQL.odbc,iMuestraMsg);
```

Use of Uninitialized Variable\Path 13:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=59
Status	New
Detection Date	12/20/2023 10:49:32 AM

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.h	DlgReporteDoctoSeguroAfirme.cpp
Line	49	939
Object	lEmpleado	lEmpleado

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.h
Method long lFacturaG,lFactura1,lFactura2,lCliente,lFolioSeguro,lEmpleado;

```
....
49.  long lFacturaG,lFactura1,lFactura2,lCliente,lFolioSeguro,lEmpleado;
```

File Name DlgReporteDoctoSeguroAfirme.cpp

Method bool CDlgReporteDoctoSeguroAfirme::transaccionGrabarDocumentosSeguro()

```
....
939.  grabarMensajeError( "C", iCaja, cIPDestino,
"ReporteDoctoSeguroAfirme", "CDlgReporteDoctoSeguroAfirme",
"transaccionGrabarDocumentosSeguro", cConsulta,lEmpleado,"ERROR
#3132",grabarDocumentoSeguro2SQL.odbc,iMuestraMsg);
```

Use of Uninitialized Variable\Path 14:

Severity	Medium
Result State	To Verify

Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=60
Status	New
Detection Date	12/20/2023 10:49:32 AM

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.h	DlgReporteDoctoSeguroAfirme.cpp
Line	50	970
Object	iMuestraMsg	iMuestraMsg

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.h
Method int iCaja,iMuestraMsg, iDiaActual,iMesActual,iAnioActual,iCiudad,

```
....
50.  int iCaja,iMuestraMsg, iDiaActual,iMesActual,iAnioActual,iCiudad,
```

File Name DlgReporteDoctoSeguroAfirme.cpp
Method long CDlgReporteDoctoSeguroAfirme::obtenerFolio(int iTipoFolio,int iIncrementarFolio)

```
....
970.  grabarMensajeError( "M", iCaja, cIPDestino,
"ReporteDoctoSeguroAfirme", "CDlgReporteDoctoSeguroAfirme",
"obtenerFolio", cConsulta,lEmpleado,"Error
#3133",folioCoppelSQL.odbc,iMuestraMsg);
```

Use of Uninitialized Variable\Path 15:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=61
Status	New
Detection Date	12/20/2023 10:49:32 AM

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.h	DlgReporteDoctoSeguroAfirme.cpp
Line	50	800
Object	iMuestraMsg	iMuestraMsg

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.h
Method int iCaja,iMuestraMsg, iDiaActual,iMesActual,iAnioActual,iCiudad,

```
....
50.  int iCaja,iMuestraMsg, iDiaActual,iMesActual,iAnioActual,iCiudad,
```

File Name DlgReporteDoctoSeguroAfirme.cpp

Method	void CDlgReporteDoctoSeguroAfirme::grabarDocumentosSeguro(long lCliente, long lFolioSeguro)//Checar para que se usaba el pcuentas_cliente->registro_cliente

```
....
800.  grabarMensajeError( "C", iCaja, cIPDestino,
"ReporteDoctoSeguroAfirme", "CDlgReporteDoctoSeguroAfirme",
"grabarDocumentoSeguro", cConsulta,lEmpleado,"Error
#3128",grabarDocumentoSeguroSQL.odbc,iMuestraMsg);
```

Use of Uninitialized Variable\Path 16:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=62
Status	New
Detection Date	12/20/2023 10:49:32 AM

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.h	DlgReporteDoctoSeguroAfirme.cpp
Line	50	844
Object	iMuestraMsg	iMuestraMsg

Code Snippet	
File Name	DlgReporteDoctoSeguroAfirme.h
Method	int iCaja,iMuestraMsg, iDiaActual,iMesActual,iAnioActual,iCiudad,

```
....
50.  int iCaja,iMuestraMsg, iDiaActual,iMesActual,iAnioActual,iCiudad,
```

File Name	DlgReporteDoctoSeguroAfirme.cpp
Method	void CDlgReporteDoctoSeguroAfirme::grabarDocumentosSeguro(long lCliente, long lFolioSeguro)//Checar para que se usaba el pcuentas_cliente->registro_cliente

```
....
844.  grabarMensajeError( "M", iCaja, cIPDestino,
"EntradaDevolucionCobranza", "CDlgEntradaDevolucionCobranza", "altaUdi",
cConsulta,lEmpleado,"Error
#3130",grabarDocumentoSeguro2SQL.odbc,iMuestraMsg);
```

Use of Uninitialized Variable\Path 17:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=63
Status	New
Detection Date	12/20/2023 10:49:32 AM

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.h	DlgReporteDoctoSeguroAfirme.cpp
Line	50	909

Object	iMuestraMsg	iMuestraMsg

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.h
Method int iCaja,iMuestraMsg, iDiaActual,iMesActual,iAnioActual,iCiudad,

```
....
50.  int iCaja,iMuestraMsg, iDiaActual,iMesActual,iAnioActual,iCiudad,
```

File Name DlgReporteDoctoSeguroAfirme.cpp

Method void CDlgReporteDoctoSeguroAfirme::grabarDocumentosSeguro(long lCliente, long lFolioSeguro)//Checar para que se usaba el pcuentas_cliente->registro_cliente

```
....
909.  grabarMensajeError( "C", iCaja, cIPDestino,
"ReporteDoctoSeguroAfirme", "CDlgReporteDoctoSeguroAfirme",
"grabarDocumentoSeguro", cConsulta,lEmpleado,"Error
#3131",grabarDocumentoSeguro2SQL.odbc,iMuestraMsg);
```

Use of Uninitialized Variable\Path 18:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=64
Status	New
Detection Date	12/20/2023 10:49:32 AM

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.h	DlgReporteDoctoSeguroAfirme.cpp
Line	50	939
Object	iMuestraMsg	iMuestraMsg

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.h
Method int iCaja,iMuestraMsg, iDiaActual,iMesActual,iAnioActual,iCiudad,

```
....
50.  int iCaja,iMuestraMsg, iDiaActual,iMesActual,iAnioActual,iCiudad,
```

File Name DlgReporteDoctoSeguroAfirme.cpp

Method bool CDlgReporteDoctoSeguroAfirme::transaccionGrabarDocumentosSeguro()

```
....
939.  grabarMensajeError( "C", iCaja, cIPDestino,
"ReporteDoctoSeguroAfirme", "CDlgReporteDoctoSeguroAfirme",
"transaccionGrabarDocumentosSeguro", cConsulta,lEmpleado,"ERROR
#3132",grabarDocumentoSeguro2SQL.odbc,iMuestraMsg);
```

Use of Uninitialized Variable\Path 19:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=65
Status	New
Detection Date	12/20/2023 10:49:32 AM

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.h	DlgReporteDoctoSeguroAfirme.cpp
Line	47	294
Object	iControles	iControles

Code Snippet

File Name DlgReporteDoctoSeguroAfirme.h

Method int iControles,iFoco;

```
....
47.  int iControles,iFoco;
```

File Name DlgReporteDoctoSeguroAfirme.cpp

Method BOOL CDlgReporteDoctoSeguroAfirme::PreTranslateMessage(MSG* pMsg)

```
....
294.  if ( iFoco <= iControles) //numero de controles
```

Use of Uninitialized Variable\Path 20:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=66
Status	New
Detection Date	12/20/2023 10:49:32 AM

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.h	DlgReporteDoctoSeguroAfirme.cpp
Line	47	392
Object	iControles	iControles

Code Snippet

File Name DlgReporteDoctoSeguroAfirme.h

Method int iControles,iFoco;

```
....
47.  int iControles,iFoco;
```

File Name DlgReporteDoctoSeguroAfirme.cpp

Method	bool CDlgReporteDoctoSeguroAfirme::validarClick(int nTmpFocus)

```
....
392.  if (iFoco != iControles)//numero de controles para los que se
quieren se vea el mensaje de error
```

Use of Uninitialized Variable\Path 21:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=67
Status	New
Detection Date	12/20/2023 10:49:32 AM

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.h	DlgReporteDoctoSeguroAfirme.cpp
Line	47	544
Object	iControles	iControles

Code Snippet	
File Name	DlgReporteDoctoSeguroAfirme.h
Method	int iControles,iFoco;

```
....
47.  int iControles,iFoco;
```

File Name	DlgReporteDoctoSeguroAfirme.cpp
Method	bool CDlgReporteDoctoSeguroAfirme::validarControles(void)

```
....
544.  for (i=0; i < iControles; i++)
```

Use of Uninitialized Variable\Path 22:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=68
Status	New
Detection Date	12/20/2023 10:49:32 AM

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.h	DlgReporteDoctoSeguroAfirme.cpp
Line	49	505
Object	lFactura1	lFactura1

Code Snippet	
File Name	DlgReporteDoctoSeguroAfirme.h
Method	long lFacturaG,lFactura1,lFactura2,lCliente,lFolioSeguro,lEmpleado;

```
....
49.  long lFacturaG,lFactura1,lFactura2,lCliente,lFolioSeguro,lEmpleado;
```

File Name DlgReporteDoctoSeguroAfirme.cpp

Method bool CDlgReporteDoctoSeguroAfirme::validarControl(char *cCadena)

```
....
505.  else if ( lFolioSeguro == lFactura1 )
```

Use of Uninitialized Variable\Path 23:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=69
Status	New
Detection Date	12/20/2023 10:49:32 AM

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.h	DlgReporteDoctoSeguroAfirme.cpp
Line	49	507
Object	lFactura1	lFactura1

Code Snippet

File Name DlgReporteDoctoSeguroAfirme.h

Method long lFacturaG,lFactura1,lFactura2,lCliente,lFolioSeguro,lEmpleado;

```
....
49.  long lFacturaG,lFactura1,lFactura2,lCliente,lFolioSeguro,lEmpleado;
```

File Name DlgReporteDoctoSeguroAfirme.cpp

Method bool CDlgReporteDoctoSeguroAfirme::validarControl(char *cCadena)

```
....
507.  lFacturaG = lFactura1;
```

Use of Uninitialized Variable\Path 24:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=70
Status	New
Detection Date	12/20/2023 10:49:32 AM

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.h	DlgReporteDoctoSeguroAfirme.cpp
Line	49	510

Object	lFactura2	lFactura2

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.h
Method long lFacturaG,lFactura1,lFactura2,lCliente,lFolioSeguro,lEmpleado;

```
....
49.  long lFacturaG,lFactura1,lFactura2,lCliente,lFolioSeguro,lEmpleado;
```

File Name DlgReporteDoctoSeguroAfirme.cpp
Method bool CDlgReporteDoctoSeguroAfirme::validarControl(char *cCadena)

```
....
510.  else if ( lFolioSeguro == lFactura2 )
```

Use of Uninitialized Variable\Path 25:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=71
Status	New
Detection Date	12/20/2023 10:49:32 AM

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.h	DlgReporteDoctoSeguroAfirme.cpp
Line	49	512
Object	lFactura2	lFactura2

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.h
Method long lFacturaG,lFactura1,lFactura2,lCliente,lFolioSeguro,lEmpleado;

```
....
49.  long lFacturaG,lFactura1,lFactura2,lCliente,lFolioSeguro,lEmpleado;
```

File Name DlgReporteDoctoSeguroAfirme.cpp
Method bool CDlgReporteDoctoSeguroAfirme::validarControl(char *cCadena)

```
....
512.  lFacturaG = lFactura2;
```

Use of Uninitialized Variable\Path 26:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=72
Status	New
Detection Date	12/20/2023 10:49:32 AM

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.h	DlgReporteDoctoSeguroAfirme.cpp
Line	49	634
Object	lCliente	lCliente

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.h
Method long lFacturaG,lFactura1,lFactura2,lCliente,lFolioSeguro,lEmpleado;

```
....
49.  long lFacturaG,lFactura1,lFactura2,lCliente,lFolioSeguro,lEmpleado;
```

File Name DlgReporteDoctoSeguroAfirme.cpp

Method void CDlgReporteDoctoSeguroAfirme::OnBnClickedOk()

```
....
634.  imprimeReciboDocumentos( lFolioSeguro, lCliente );
```

Use of Uninitialized Variable\Path 27:

Severity Medium
Result State To Verify
Online Results https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=73
Status New
Detection Date 12/20/2023 10:49:32 AM

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.h	DlgReporteDoctoSeguroAfirme.cpp
Line	51	655
Object	iFinLinea	iFinLinea

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.h
Method iFinLinea,iTipoImpresora;

```
....
51.  iFinLinea,iTipoImpresora;
```

File Name DlgReporteDoctoSeguroAfirme.cpp

Method void CDlgReporteDoctoSeguroAfirme::imprimeReciboDocumentos(long lFolioSeguro, long lCliente)

```
....
655.  C_FormasPCL hoja(33,270,"LPT1",iFinLinea, iTipoImpresora );
```

Use of Uninitialized Variable\Path 28:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=74
Status	New
Detection Date	12/20/2023 10:49:32 AM

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.h	DlgReporteDoctoSeguroAfirme.cpp
Line	51	655
Object	iTipoImpresora	iTipoImpresora

Code Snippet

File Name DlgReporteDoctoSeguroAfirme.h

Method iFinLinea,iTipoImpresora;

```
....
51.  iFinLinea,iTipoImpresora;
```

File Name DlgReporteDoctoSeguroAfirme.cpp

Method void CDlgReporteDoctoSeguroAfirme::imprimeReciboDocumentos(long lFolioSeguro, long lCliente)

```
....
655.  C_FormasPCL hoja(33,270,"LPT1",iFinLinea, iTipoImpresora );
```

Use of Uninitialized Variable\Path 29:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=75
Status	New
Detection Date	12/20/2023 10:49:32 AM

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.h	DlgReporteDoctoSeguroAfirme.cpp
Line	50	879
Object	iCiudad	iCiudad

Code Snippet

File Name DlgReporteDoctoSeguroAfirme.h

Method int iCaja,iMuestraMsg, iDiaActual,iMesActual,iAnioActual,iCiudad,

```
....
50.  int iCaja,iMuestraMsg, iDiaActual,iMesActual,iAnioActual,iCiudad,
```

File Name DlgReporteDoctoSeguroAfirme.cpp

Method	void CDlgReporteDoctoSeguroAfirme::grabarDocumentosSeguro(long lCliente, long lFolioSeguro)//Checar para que se usaba el pcuentas_cliente->registro_cliente

```
....
879.  grabarDocumentoSeguro2SQL.ciudad = short int(iCiudad);
```

Use of Uninitialized Variable\Path 30:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=76
Status	New
Detection Date	12/20/2023 10:49:32 AM

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.h	DlgReporteDoctoSeguroAfirme.cpp
Line	50	882
Object	iDiaActual	iDiaActual

Code Snippet

File Name DlgReporteDoctoSeguroAfirme.h

Method int iCaja,iMuestraMsg, iDiaActual,iMesActual,iAnioActual,iCiudad,

```
....
50.  int iCaja,iMuestraMsg, iDiaActual,iMesActual,iAnioActual,iCiudad,
```

File Name DlgReporteDoctoSeguroAfirme.cpp

Method void CDlgReporteDoctoSeguroAfirme::grabarDocumentosSeguro(long lCliente, long lFolioSeguro)//Checar para que se usaba el pcuentas_cliente->registro_cliente

```
....
882.
grabarDocumentoSeguro2SQL.fecha.ponerFecha(iDiaActual,iMesActual,iAnioActual);
```

Use of Uninitialized Variable\Path 31:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=77
Status	New
Detection Date	12/20/2023 10:49:32 AM

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.h	DlgReporteDoctoSeguroAfirme.cpp
Line	50	882
Object	iMesActual	iMesActual

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.h
Method int iCaja,iMuestraMsg, iDiaActual,iMesActual,iAnioActual,iCiudad,

```
....
50.  int iCaja,iMuestraMsg, iDiaActual,iMesActual,iAnioActual,iCiudad,
```

File Name DlgReporteDoctoSeguroAfirme.cpp

Method void CDlgReporteDoctoSeguroAfirme::grabarDocumentosSeguro(long lCliente, long lFolioSeguro)//Checar para que se usaba el pcuentas_cliente->registro_cliente

```
....
882.
grabarDocumentoSeguro2SQL.fecha.ponerFecha(iDiaActual,iMesActual,iAnioAc
tual);
```

Use of Uninitialized Variable\Path 32:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=78
Status	New
Detection Date	12/20/2023 10:49:32 AM

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.h	DlgReporteDoctoSeguroAfirme.cpp
Line	50	882
Object	iAnioActual	iAnioActual

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.h
Method int iCaja,iMuestraMsg, iDiaActual,iMesActual,iAnioActual,iCiudad,

```
....
50.  int iCaja,iMuestraMsg, iDiaActual,iMesActual,iAnioActual,iCiudad,
```

File Name DlgReporteDoctoSeguroAfirme.cpp

Method void CDlgReporteDoctoSeguroAfirme::grabarDocumentosSeguro(long lCliente, long lFolioSeguro)//Checar para que se usaba el pcuentas_cliente->registro_cliente

```
....
882.
grabarDocumentoSeguro2SQL.fecha.ponerFecha(iDiaActual,iMesActual,iAnioAc
tual);
```

Memory Leak

Query Path:
CPP\Cx\CPP Medium Threat\Memory Leak Version:8

Categories

NIST SP 800-53: SC-5 Denial of Service Protection (P1)
ASD STIG 5.2: APSC-DV-002400 - CAT II The application must restrict the ability to launch Denial of Service (DoS) attacks against itself or other information systems.

Description
Memory Leak\Path 1:

Severity	Medium
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=46
Status	New
Detection Date	12/20/2023 10:49:32 AM

The memory allocated for CDynLinkLibrary in ReporteDoctoSeguroAfirme.cpp at line 14 is not always properly deallocated in its declaring scope, resulting in a potential memory leak.

	Source	Destination
File	ReporteDoctoSeguroAfirme.cpp	ReporteDoctoSeguroAfirme.cpp
Line	39	39
Object	CDynLinkLibrary	CDynLinkLibrary

Code Snippet
File Name ReporteDoctoSeguroAfirme.cpp
Method DllMain(HINSTANCE hInstance, DWORD dwReason, LPVOID lpReserved)

```
....
39.  new CDynLinkLibrary(ReporteDoctoSeguroAfirmeDLL);
```

Use of Obsolete Functions

Query Path:
CPP\Cx\CPP Low Visibility\Use of Obsolete Functions Version:2

Categories

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
OWASP ASVS: V01 Architecture, Design and Threat Modeling

Description
Use of Obsolete Functions\Path 1:

Severity	Low
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=92
Status	New
Detection Date	12/20/2023 10:49:32 AM

Method BEGIN_MESSAGE_MAP in DlgReporteDoctoSeguroAfirme.cpp, at line 87, calls an obsolete API, memset. This has been deprecated, and should not be used in a modern codebase.

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.cpp	DlgReporteDoctoSeguroAfirme.cpp
Line	155	155

Object	memset	memset

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.cpp
Method BEGIN_MESSAGE_MAP(CDlgReporteDoctoSeguroAfirme, CDialogoML)

```
....
155.  memset(cFecha,0,sizeof(cFecha));
```

Use of Obsolete Functions\Path 2:

Severity	Low
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=93
Status	New
Detection Date	12/20/2023 10:49:32 AM

Method BEGIN_MESSAGE_MAP in DlgReporteDoctoSeguroAfirme.cpp, at line 87, calls an obsolete API, memset. This has been deprecated, and should not be used in a modern codebase.

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.cpp	DlgReporteDoctoSeguroAfirme.cpp
Line	156	156
Object	memset	memset

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.cpp
Method BEGIN_MESSAGE_MAP(CDlgReporteDoctoSeguroAfirme, CDialogoML)

```
....
156.  memset(cNombreCiudad,0,sizeof(cNombreCiudad));
```

Use of Obsolete Functions\Path 3:

Severity	Low
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=94
Status	New
Detection Date	12/20/2023 10:49:32 AM

Method CDlgReporteDoctoSeguroAfirme::obtenerFolio in DlgReporteDoctoSeguroAfirme.cpp, at line 946, calls an obsolete API, memset. This has been deprecated, and should not be used in a modern codebase.

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.cpp	DlgReporteDoctoSeguroAfirme.cpp
Line	952	952
Object	memset	memset

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.cpp
Method long CDlgReporteDoctoSeguroAfirme::obtenerFolio(int iTipoFolio,int iIncrementarFolio)

```
....
952.  memset(cIPDestino,0,sizeof(cIPDestino));
```

Use of Obsolete Functions\Path 4:

Severity	Low
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=95
Status	New
Detection Date	12/20/2023 10:49:32 AM

Method CDlgReporteDoctoSeguroAfirme::obtenerFolio in DlgReporteDoctoSeguroAfirme.cpp, at line 946, calls an obsolete API, memset. This has been deprecated, and should not be used in a modern codebase.

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.cpp	DlgReporteDoctoSeguroAfirme.cpp
Line	953	953
Object	memset	memset

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.cpp
Method long CDlgReporteDoctoSeguroAfirme::obtenerFolio(int iTipoFolio,int iIncrementarFolio)

```
....
953.  memset(cConsulta,0,sizeof(cConsulta));
```

Use of Obsolete Functions\Path 5:

Severity	Low
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=96
Status	New
Detection Date	12/20/2023 10:49:32 AM

Method CDlgReporteDoctoSeguroAfirme::validarClick in DlgReporteDoctoSeguroAfirme.cpp, at line 377, calls an obsolete API, memset. This has been deprecated, and should not be used in a modern codebase.

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.cpp	DlgReporteDoctoSeguroAfirme.cpp
Line	384	384
Object	memset	memset

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.cpp
Method bool CDlgReporteDoctoSeguroAfirme::validarClick(int nTmpFocus)

```
....
384.  memset(cPaso,0,sizeof(cPaso));
```

Use of Obsolete Functions\Path 6:

Severity	Low

Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=97
Status	New
Detection Date	12/20/2023 10:49:32 AM

Method CDlgReporteDoctoSeguroAfirme::validarControl in DlgReporteDoctoSeguroAfirme.cpp, at line 473, calls an obsolete API, memset. This has been deprecated, and should not be used in a modern codebase.

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.cpp	DlgReporteDoctoSeguroAfirme.cpp
Line	489	489
Object	memset	memset

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.cpp
Method bool CDlgReporteDoctoSeguroAfirme::validarControl(char *cCadena)

```
....
489.  memset(cCadena,' ',80);
```

Use of Obsolete Functions\Path 7:

Severity	Low
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=98
Status	New
Detection Date	12/20/2023 10:49:32 AM

Method CDlgReporteDoctoSeguroAfirme::validarControl in DlgReporteDoctoSeguroAfirme.cpp, at line 473, calls an obsolete API, memset. This has been deprecated, and should not be used in a modern codebase.

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.cpp	DlgReporteDoctoSeguroAfirme.cpp
Line	498	498
Object	memset	memset

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.cpp
Method bool CDlgReporteDoctoSeguroAfirme::validarControl(char *cCadena)

```
....
498.  memset(cCadena,' ',80);
```

Use of Obsolete Functions\Path 8:

Severity	Low
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=99
Status	New
Detection Date	12/20/2023 10:49:32 AM

Method CDlgReporteDoctoSeguroAfirme::validarControl in DlgReporteDoctoSeguroAfirme.cpp, at line 473, calls an obsolete API, memset. This has been deprecated, and should not be used in a modern codebase.

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.cpp	DlgReporteDoctoSeguroAfirme.cpp
Line	517	517
Object	memset	memset

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.cpp
Method bool CDlgReporteDoctoSeguroAfirme::validarControl(char *cCadena)

```
....
517.  memset(cCadena,' ',80);
```

Use of Obsolete Functions\Path 9:

Severity	Low
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=100
Status	New
Detection Date	12/20/2023 10:49:32 AM

Method CDlgReporteDoctoSeguroAfirme::validarControles in DlgReporteDoctoSeguroAfirme.cpp, at line 533, calls an obsolete API, memset. This has been deprecated, and should not be used in a modern codebase.

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.cpp	DlgReporteDoctoSeguroAfirme.cpp
Line	540	540
Object	memset	memset

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.cpp
Method bool CDlgReporteDoctoSeguroAfirme::validarControles(void)

```
....
540.  memset(cPaso,0,sizeof(cPaso));
```

Use of Obsolete Functions\Path 10:

Severity	Low
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=101
Status	New
Detection Date	12/20/2023 10:49:32 AM

Method CDlgReporteDoctoSeguroAfirme::grabarDocumentosSeguro in DlgReporteDoctoSeguroAfirme.cpp, at line 773, calls an obsolete API, memset. This has been deprecated, and should not be used in a modern codebase.

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.cpp	DlgReporteDoctoSeguroAfirme.cpp
Line	782	782

Object	memset	memset

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.cpp
Method void CDlgReporteDoctoSeguroAfirme::grabarDocumentosSeguro(long lCliente, long lFolioSeguro)//Checar para que se usaba el pcuentas_cliente->registro_cliente

```
....
782.  memset(cTexto,0,10);
```

Use of Obsolete Functions\Path 11:

Severity	Low
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=102
Status	New
Detection Date	12/20/2023 10:49:32 AM

Method CDlgReporteDoctoSeguroAfirme::grabarDocumentosSeguro in DlgReporteDoctoSeguroAfirme.cpp, at line 773, calls an obsolete API, memset. This has been deprecated, and should not be used in a modern codebase.

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.cpp	DlgReporteDoctoSeguroAfirme.cpp
Line	783	783
Object	memset	memset

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.cpp
Method void CDlgReporteDoctoSeguroAfirme::grabarDocumentosSeguro(long lCliente, long lFolioSeguro)//Checar para que se usaba el pcuentas_cliente->registro_cliente

```
....
783.  memset(cRespuesta,0,10);
```

Use of Obsolete Functions\Path 12:

Severity	Low
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=103
Status	New
Detection Date	12/20/2023 10:49:32 AM

Method CA0068 in main.cpp, at line 18, calls an obsolete API, memset. This has been deprecated, and should not be used in a modern codebase.

	Source	Destination
File	main.cpp	main.cpp
Line	34	34
Object	memset	memset

Code Snippet
File Name main.cpp

Method	int CA0068(char *cInput1,char *cInput2)

```
....
34.  memset( &cParametros, 0, sizeof( SParametros ) );
```

Use of Obsolete Functions\Path 13:

Severity	Low
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=104
Status	New
Detection Date	12/20/2023 10:49:32 AM

Method CA0068 in main.cpp, at line 18, calls an obsolete API, memset. This has been deprecated, and should not be used in a modern codebase.

	Source	Destination
File	main.cpp	main.cpp
Line	40	40
Object	memset	memset

Code Snippet	
File Name	main.cpp
Method	int CA0068(char *cInput1,char *cInput2)

```
....
40.  memset( cServer,0,sizeof(cServer) );
```

Use of Obsolete Functions\Path 14:

Severity	Low
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=105
Status	New
Detection Date	12/20/2023 10:49:32 AM

Method CDlgReporteDoctoSeguroAfirme::validarControl in DlgReporteDoctoSeguroAfirme.cpp, at line 473, calls an obsolete API, atol. This has been deprecated, and should not be used in a modern codebase.

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.cpp	DlgReporteDoctoSeguroAfirme.cpp
Line	485	485
Object	atol	atol

Code Snippet	
File Name	DlgReporteDoctoSeguroAfirme.cpp
Method	bool CDlgReporteDoctoSeguroAfirme::validarControl(char *cCadena)

```
....
485.  lFolioSeguro = atol(sFolioSeguro);
```

Use of Obsolete Functions\Path 15:

Severity	Low
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=106
Status	New
Detection Date	12/20/2023 10:49:32 AM

Method CA0068 in main.cpp, at line 18, calls an obsolete API, atol. This has been deprecated, and should not be used in a modern codebase.

	Source	Destination
File	main.cpp	main.cpp
Line	66	66
Object	atol	atol

Code Snippet
File Name main.cpp
Method int CA0068(char *cInput1,char *cInput2)

```
....
66.  dlg.lCliente = atol(sCliente);
```

Use of Obsolete Functions\Path 16:

Severity	Low
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=107
Status	New
Detection Date	12/20/2023 10:49:32 AM

Method CA0068 in main.cpp, at line 18, calls an obsolete API, atol. This has been deprecated, and should not be used in a modern codebase.

	Source	Destination
File	main.cpp	main.cpp
Line	69	69
Object	atol	atol

Code Snippet
File Name main.cpp
Method int CA0068(char *cInput1,char *cInput2)

```
....
69.  dlg.lFactura1 = atol(sFactura1);
```

Use of Obsolete Functions\Path 17:

Severity	Low
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=108
Status	New
Detection Date	12/20/2023 10:49:32 AM

Method CA0068 in main.cpp, at line 18, calls an obsolete API, atol. This has been deprecated, and should not be used in a modern codebase.

	Source	Destination
File	main.cpp	main.cpp
Line	72	72
Object	atol	atol

Code Snippet
File Name main.cpp
Method int CA0068(char *cInput1,char *cInput2)

```
....
72.  dlg.lFactura2 = atol(sFactura2);
```

Unchecked Return Value

Query Path:
CPP\Cx\CPP Low Visibility\Unchecked Return Value Version:1

Categories

NIST SP 800-53: SI-11 Error Handling (P2)
ASD STIG 5.2: APSC-DV-002580 - CAT II The application must reveal error messages only to the ISSO, ISSM, or SA.
PCI DSS v4.0: PCI DSS (4.0) - 6.2.4 Vulnerabilities in software development

Description
Unchecked Return Value\Path 1:

Severity	Low
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=79
Status	New
Detection Date	12/20/2023 10:49:32 AM

The CDlgReporteDoctoSeguroAfirme::obtenerFolio method calls the sprintf function, at line 946 of DlgReporteDoctoSeguroAfirme.cpp. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.cpp	DlgReporteDoctoSeguroAfirme.cpp
Line	963	963
Object	sprintf	sprintf

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.cpp
Method long CDlgReporteDoctoSeguroAfirme::obtenerFolio(int iTipoFolio,int iIncrementarFolio)

```
....
963.  sprintf(cIPDestino,"%s",sServer);
```

Unchecked Return Value\Path 2:

Severity	Low
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185

	&pathid=80
Status	New
Detection Date	12/20/2023 10:49:32 AM

The CDlgReporteDoctoSeguroAfirme::obtenerFolio method calls the sprintf function, at line 946 of DlgReporteDoctoSeguroAfirme.cpp. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.cpp	DlgReporteDoctoSeguroAfirme.cpp
Line	966	966
Object	sprintf	sprintf

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.cpp
Method long CDlgReporteDoctoSeguroAfirme::obtenerFolio(int iTipoFolio,int iIncrementarFolio)

```
....
966.   sprintf(cConsulta,"%s",sSqlTxt);
```

Unchecked Return Value\Path 3:

Severity	Low
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=81
Status	New
Detection Date	12/20/2023 10:49:32 AM

The CDlgReporteDoctoSeguroAfirme::grabarDocumentosSeguro method calls the sprintf function, at line 773 of DlgReporteDoctoSeguroAfirme.cpp. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.cpp	DlgReporteDoctoSeguroAfirme.cpp
Line	793	793
Object	sprintf	sprintf

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.cpp
Method void CDlgReporteDoctoSeguroAfirme::grabarDocumentosSeguro(long lCliente, long lFolioSeguro)//Checar para que se usaba el pcuentas_cliente->registro_cliente

```
....
793.   sprintf(cIPDestino,"%s",sServer);
```

Unchecked Return Value\Path 4:

Severity	Low
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=82
Status	New
Detection Date	12/20/2023 10:49:32 AM

The CDlgReporteDoctoSeguroAfirme::grabarDocumentosSeguro method calls the sprintf function, at line 773 of DlgReporteDoctoSeguroAfirme.cpp. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.cpp	DlgReporteDoctoSeguroAfirme.cpp
Line	796	796
Object	sprintf	sprintf

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.cpp
Method void CDlgReporteDoctoSeguroAfirme::grabarDocumentosSeguro(long lCliente, long lFolioSeguro)//Checar para que se usaba el pcuentas_cliente->registro_cliente

```
....
796.  sprintf(cConsulta,"%s",sSqlTxt);
```

Unchecked Return Value\Path 5:

Severity	Low
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=83
Status	New
Detection Date	12/20/2023 10:49:32 AM

The CDlgReporteDoctoSeguroAfirme::grabarDocumentosSeguro method calls the sprintf function, at line 773 of DlgReporteDoctoSeguroAfirme.cpp. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.cpp	DlgReporteDoctoSeguroAfirme.cpp
Line	837	837
Object	sprintf	sprintf

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.cpp
Method void CDlgReporteDoctoSeguroAfirme::grabarDocumentosSeguro(long lCliente, long lFolioSeguro)//Checar para que se usaba el pcuentas_cliente->registro_cliente

```
....
837.  sprintf(cIPDestino,"%s",sServer);
```

Unchecked Return Value\Path 6:

Severity	Low
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=84
Status	New
Detection Date	12/20/2023 10:49:32 AM

The CDlgReporteDoctoSeguroAfirme::grabarDocumentosSeguro method calls the sprintf function, at line 773 of DlgReporteDoctoSeguroAfirme.cpp. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.cpp	DlgReporteDoctoSeguroAfirme.cpp
Line	840	840
Object	sprintf	sprintf

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.cpp
Method void CDlgReporteDoctoSeguroAfirme::grabarDocumentosSeguro(long lCliente, long lFolioSeguro)//Checar para que se usaba el pcuentas_cliente->registro_cliente

```
....
840.  sprintf(cConsulta,"%s",sSqlTxt);
```

Unchecked Return Value\Path 7:

Severity	Low
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=85
Status	New
Detection Date	12/20/2023 10:49:32 AM

The CDlgReporteDoctoSeguroAfirme::grabarDocumentosSeguro method calls the sprintf function, at line 773 of DlgReporteDoctoSeguroAfirme.cpp. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.cpp	DlgReporteDoctoSeguroAfirme.cpp
Line	891	891
Object	sprintf	sprintf

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.cpp
Method void CDlgReporteDoctoSeguroAfirme::grabarDocumentosSeguro(long lCliente, long lFolioSeguro)//Checar para que se usaba el pcuentas_cliente->registro_cliente

```
....
891.  sprintf(cIPCacarmov,"%s",sServer);
```

Unchecked Return Value\Path 8:

Severity	Low
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=86
Status	New
Detection Date	12/20/2023 10:49:32 AM

The CDlgReporteDoctoSeguroAfirme::grabarDocumentosSeguro method calls the sprintf function, at line 773 of DlgReporteDoctoSeguroAfirme.cpp. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.cpp	DlgReporteDoctoSeguroAfirme.cpp

Line	902	902
Object	sprintf	sprintf

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.cpp
Method void CDlgReporteDoctoSeguroAfirme::grabarDocumentosSeguro(long lCliente, long lFolioSeguro)//Checar para que se usaba el pcuentas_cliente->registro_cliente

```
....
902.  sprintf(cIPDestino,"%s",sServer);
```

Unchecked Return Value\Path 9:

Severity	Low
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=87
Status	New
Detection Date	12/20/2023 10:49:32 AM

The CDlgReporteDoctoSeguroAfirme::grabarDocumentosSeguro method calls the sprintf function, at line 773 of DlgReporteDoctoSeguroAfirme.cpp. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.cpp	DlgReporteDoctoSeguroAfirme.cpp
Line	905	905
Object	sprintf	sprintf

Code Snippet
File Name DlgReporteDoctoSeguroAfirme.cpp
Method void CDlgReporteDoctoSeguroAfirme::grabarDocumentosSeguro(long lCliente, long lFolioSeguro)//Checar para que se usaba el pcuentas_cliente->registro_cliente

```
....
905.  sprintf(cConsulta,"%s",sSqlTxt);
```

Unchecked Return Value\Path 10:

Severity	Low
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=88
Status	New
Detection Date	12/20/2023 10:49:32 AM

The CDlgReporteDoctoSeguroAfirme::transaccionGrabarDocumentosSeguro method calls the sprintf function, at line 918 of DlgReporteDoctoSeguroAfirme.cpp. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.cpp	DlgReporteDoctoSeguroAfirme.cpp
Line	932	932
Object	sprintf	sprintf

Code Snippet	
File Name	DlgReporteDoctoSeguroAfirme.cpp
Method	bool CDlgReporteDoctoSeguroAfirme::transaccionGrabarDocumentosSeguro()

```
....
932.  sprintf(cIPDestino,"%s",sServer);
```

Unchecked Return Value\Path 11:

Severity	Low
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=89
Status	New
Detection Date	12/20/2023 10:49:32 AM

The CDlgReporteDoctoSeguroAfirme::transaccionGrabarDocumentosSeguro method calls the sprintf function, at line 918 of DlgReporteDoctoSeguroAfirme.cpp. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	DlgReporteDoctoSeguroAfirme.cpp	DlgReporteDoctoSeguroAfirme.cpp
Line	935	935
Object	sprintf	sprintf

Code Snippet	
File Name	DlgReporteDoctoSeguroAfirme.cpp
Method	bool CDlgReporteDoctoSeguroAfirme::transaccionGrabarDocumentosSeguro()

```
....
935.  sprintf(cConsulta,"%s",sSqlTxt);
```

Unchecked Return Value\Path 12:

Severity	Low
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=90
Status	New
Detection Date	12/20/2023 10:49:32 AM

The CA0068 method calls the sprintf function, at line 18 of main.cpp. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	main.cpp	main.cpp
Line	47	47
Object	sprintf	sprintf

Code Snippet	
File Name	main.cpp
Method	int CA0068(char *cInput1,char *cInput2)

```
....
47.  sprintf(cIpTiendaLocal, "%s", sServer);
```

Unchecked Return Value\Path 13:

Severity	Low
Result State	To Verify
Online Results	https://coppel.checkmarx.net/CxWebClient/ViewerMain.aspx?scanid=1340630&projectid=79185&pathid=91
Status	New
Detection Date	12/20/2023 10:49:32 AM

The CA0068 method calls the sprintf function, at line 18 of main.cpp. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	main.cpp	main.cpp
Line	48	48
Object	sprintf	sprintf

Code Snippet
File Name main.cpp
Method int CA0068(char *cInput1,char *cInput2)

```
....
48.  sprintf(cServer, "%s", sServer);
```

Buffer Overflow Unbounded Format

Risk

What might happen

Failure to restrict memory writes into an intended buffer may allow attackers to write arbitrary data into memory by overflowing the destination buffer with malicious data, corrupting the memory, resulting in unpredictable behavior and, potentially, in execution of malicious code.

Failure to restrict memory writes into a format string, which in turn is written to a buffer, may allow attackers to write arbitrary data into memory by overflowing the destination buffer with malicious data, corrupting the memory, resulting in unpredictable behavior and, potentially, in execution of malicious code.

Cause

How does it happen

A Buffer Overflow occurs when code attempts to perform raw memory writes to an allocated memory buffer, but fails to ensure writes cannot exceed, "miss" or escape the given buffer. If this can be exploited, a mismanaged memory write could end up writing data beyond its allocated buffer, corrupting memory. If this memory corruption can be initiated, controlled or manipulated by an attacker, they may corrupt specific data, divert application logic flow, and execute malicious code.

General Recommendations

How to avoid it

- Always perform strict bounds checks, where required, before writing to a format string, in order to ensure data is not written beyond its intended destination buffer bounds.
- Where possible, consider using alternative functions that either contain their own internal bounds checks, or require bounds to be provided as part of the function parameters. If bounds are provided as part of function parameters, ensure they are properly deduced from the source buffer - incorrect bounds are also likely to result in a buffer overflow.

Source Code Examples

CPP

Using sprintf To Create a Format String

```
int LINE_LENGTH = 100;

std::string addLineNumber(int num, std::string msg) {
      char buf[LINE_LENGTH];
      sprintf(buf, "%d - %s", num, msg.c_str()); // if length of num and size of msg exceed
100, a buffer overflow will occur
      return std::string(buf);
}
```

Using snprintf To Create a Format String, with The Destination Buffer Determining the Amount of Bytes to be Copied

```
int LINE_LENGTH = 100;

std::string addLineNumber_fixed(int num, std::string msg) {
      char buf[LINE_LENGTH];
      snprintf(buf, sizeof(buf),"%d - %s", num, msg.c_str()); // sizeof(buf) ensures the
source buffer is not larger than the destination buffer
      return std::string(buf);
}
```

Using scanf function to copy user input to a variable

```
using namespace std;

string scanStr() {
    char buf[100];
    scanf("%s", buf);       // Buffer Overflow will occur if the string passed from STDIN
with scanf is larger than buf

    return string(buf);
}
```

Dangerous Functions

Risk

What might happen

Use of dangerous functions may expose varying risks associated with each particular function, with potential impact of improper usage of these functions varying significantly. The presence of such functions indicates a flaw in code maintenance policies and adherence to secure coding practices, in a way that has allowed introducing known dangerous code into the application.

Cause

How does it happen

A dangerous function has been identified within the code. Functions are often deemed dangerous to use for numerous reasons, as there are different sets of vulnerabilities associated with usage of such functions. For example, some string copy and concatenation functions are vulnerable to Buffer Overflow, Memory Disclosure, Denial of Service and more. Use of these functions is not recommended.

General Recommendations

How to avoid it

- Deploy a secure and recommended alternative to any functions that were identified as dangerous.
    - If no secure alternative is found, conduct further researching and testing to identify whether current usage successfully sanitizes and verifies values, and thus successfully avoids the use-cases for whom the function is indeed dangerous
- Conduct a periodical review of methods that are in use, to ensure that all external libraries and built-in functions are up-to-date and whose use has not been excluded from best secure coding practices.

Source Code Examples

CPP

Buffer Overflow in gets()

```
int main()
{
	char buf[10];

	printf("Please enter your name: ");
	gets(buf); // veryveryverylongname
	if (buf == ACCEPTED_NAME)
	{
		// Do something
	}
	return 0;
}
```

Safe reading from user

```
int main()
{
	char buf[10];

	printf("Please enter your name: ");
	fgets(buf, sizeof(buf), stdin); //setting the amount of bytes to read
	if (buf == ACCEPTED_NAME)
```

```
	{
		//Do something
	}
	return 0;
}
```

Unsafe function for string copy

```
int main(int argc, char* argv[])
{
	char buf[10];
	strcpy(buf, argv[1]); // overflow occurs when len(argv[1]) > 10 bytes

	return 0;
}
```

Safe string copy

```
int main(int argc, char* argv[])
{
	char buf[10];
	strncpy(buf, argv[1], sizeof(buf));
	buf[9]= '\0'; //strncpy doesn't NULL terminates

	return 0;
}
```

Unsafe format string

```
int main(int argc, char* argv[])
{
	printf(argv[1]); // If argv[1] contains a format token, such as %s,%x or %d, will cause
an access violation
	return 0;
}
```

Safe format string

```
int main(int argc, char* argv[])
{
	printf("%s", argv[1]); // Second parameter is not a formattable string

	return 0;
}
```

Memory Leak

Risk

What might happen

If a memory leak occurs anywhere in code in a way that is unpredictable, an application may bloat over a period of time until it exhausts all available memory, at which point slowdown and even crashes are likely to occur. However, if an attacker can induce a memory leak, they may use it to intentionally cause denial-of-service.

Cause

How does it happen

Memory leaks occur when memory is allocated in a way which would require release, but is not properly released after use. Practically speaking, this often occurs when erroneous conditions skip the memory release phase, or when it is not determined where exactly in code memory is released, resulting in a situation where it is not properly released anywhere.

General Recommendations

How to avoid it

- Ensure memory is always released in the same scope it is declared, to avoid situations where memory release is neglected.
- When handling errors, always ensure all potential code flows neatly discard all allocated memory - even when errors occur.

Source Code Examples

CPP

Memory Leak via Memory Allocation without Free

```
void someFunction(){
    int * myvar = malloc(sizeof(int)); /* myvar is never released with free(), resulting in
every call to

                                                    someFunction assigning heap
memory without releasing it, leaking memory */
}
```

Memory Leak via Reassigned Pointer to Memory Allocation

```
void someFunction(){
    int * myvar = malloc(sizeof(int)); //allocate memory
    /* Do something */
      myvar = 0; // myvar is reassigned
      /* Do something */
      free(myvar); // the original malloc is never released
}
```

Memory Leak via Object Creation without Delete

```
void someFunction(){
    char * myvar = new char[10]; /* myvar is never released with delete, resulting in every
call

                                                    to someFunction assigning heap memory
```

```
without releasing it, leaking memory*/
}
```

Freeing Allocated Memory to Avoid Memory Leaks

```
void someFunction(){
    int * myvar = malloc(sizeof(int));
      /* Do something */
      free(myvar);
}
```

Deleting New Objects to Avoid Memory Leaks

```
void someFunction(){
    char * myvar = new char[10];
      /* Do something */
    delete[] myvar;
}
```

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

Unchecked Return Value

Risk

What might happen

A program that does not check function return values could cause the application to enter an undefined state. This could lead to unexpected behavior and unintended consequences, including inconsistent data, system crashes or other error-based exploits.

Cause

How does it happen

The application calls a function, but does not check the result of this function's return values. The application simply ignores the result value, using it or passing it on with ensuring it is correct and desired, first.

General Recommendations

How to avoid it

- Always check the result of any called function that returns a value, and verify the result is an expected value.
- Ensure the calling function responds to all possible return values.
- Expect runtime errors and handle them gracefully. Explicitly define a mechanism for handling unexpected errors.

Source Code Examples

CPP

Unchecked Memory Allocation - If Malloc Fails, a NULL is Returned

```
buff = (char*) malloc(size);
strncpy(buff, source, size-1); // Assume source and size are validated prior
buff[size] = '\0';
```

Check for Successful Memory Allocation

```
buff = (char*) malloc(size);
if (buff == NULL) {
  exit(OUT_OF_MEMORY_EXIT_CODE); // Program ran out of memory or size was -1; exiting is one
way to ensure the program doesn't crash unexpectedly
}
strncpy(buff, source, size-1); // Assume source and size are validated prior
buff[size] = '\0';
```

Use of Obsolete Functions

Risk

What might happen

Referencing deprecated modules can cause an application to be exposed to known vulnerabilities, that have been publicly reported and already fixed. A common attack technique is to scan applications for these known vulnerabilities, and then exploit the application through these deprecated versions. However, even if deprecated code is used in a way that is completely secure, its very use and inclusion in the code base would encourage developers to re-use the deprecated element in the future, potentially leaving the application vulnerable to attack, which is why deprecated code should be eliminated from the code-base as a matter of practice. Note that the actual risk involved depends on the specifics of any known vulnerabilities in older versions.

Cause

How does it happen

The application references code elements that have been declared as deprecated. This could include classes, functions, methods, properties, modules, or obsolete library versions that are either out of date by version, or have been entirely deprecated. It is likely that the code that references the obsolete element was developed before it was declared as obsolete, and in the meantime the referenced code was updated.

General Recommendations

How to avoid it

- Always prefer to use the most updated versions of libraries, packages, and other dependancies.
- Do not use or reference any class, method, function, property, or other element that has been declared deprecated.

Source Code Examples

Java

Using Deprecated Methods for Security Checks

```
private void checkPermissions(InetAddress address) {

    SecurityManager secManager = System.getSecurityManager();

    if (secManager != null) {
        secManager.checkMulticast(address, 0)
    }

}
```

A Replacement Security Check

```
private void checkPermissions(InetAddress address) {

    SecurityManager secManager = System.getSecurityManager();

    if (secManager != null) {
        SocketPermission permission = new SocketPermission(address.getHostAddress(),
"accept,connect");

        secManager.checkPermission(permission)
    }

}
```

Scanned Languages

Language	Hash Number	Change Date
CPP	5434069464174654	10/29/2023
Common	1748526439669575	10/29/2023