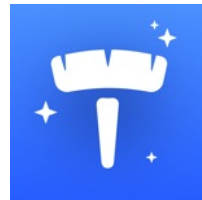




## IOS STATIC ANALYSIS REPORT





### 🍏 Turbo Cleaner (1.3.0)

File Name:	com.turbocleaner.aiclean_6738571716_1.3.0.ipa
Identifier:	com.turbocleaner.aiclean
Scan Date:	Jan. 11, 2025, 3:05 p.m.
App Security Score:	<b>60/100 (LOW RISK)</b>

Grade:



## FINDINGS SEVERITY

 HIGH	 MEDIUM	 INFO	 SECURE	 HOTSPOT
0	6	1	1	1

## FILE INFORMATION

**File Name:** com.turbocleaner.aiclean\_6738571716\_1.3.0.ipa

**Size:** 46.89MB

**MDS:** 367cebcb709c5e8c21637b6d9ecde8c8

**SHA1:** df0ddf9fa335f1ce81c6230c6d7021b3e18ea88b

**SHA256:** d43b288b1e32bb1ad96f6a6d6e964209d870aab34af351380615d13eac38a9ca

## APP INFORMATION

**App Name:** Turbo Cleaner

**App Type:** Swift

**Identifier:** com.turbocleaner.aiclean

SDK Name: iphoneos18.0  
Version: 1.3.0  
Build: 100300  
Platform Version: 18.0  
Min OS Version: 15.0  
Supported Platforms: iPhoneOS,

## BINARY INFORMATION

Arch: ARM64  
Sub Arch: CPU\_SUBTYPE\_ARM64\_ALL  
Bit: 64-bit  
Endian: <

## #CUSTOM URL SCHEMES

URL NAME	SCHEMES
None	fb1155134272808368

## APPLICATION PERMISSIONS

PERMISSIONS	STATUS	INFO	REASON IN MANIFEST
NSCalendarsUsageDescription	dangerous	Access Calendars.	App needs access to calendar to manage events
NSContactsUsageDescription	dangerous	Access Contacts.	Get contacts permission to modify contact information
NSPhotoLibraryUsageDescription	dangerous	Access the user's photo library.	This allows your excess photos and videos to be found to free up storage

## APP TRANSPORT SECURITY (ATS)

NO	ISSUE	SEVERITY	DESCRIPTION
----	-------	----------	-------------

## IPA BINARY CODE ANALYSIS

HIGH: 0 | WARNING: 3 | INFO: 1 | SECURE: 0 | SUPPRESSED: 0

NO	ISSUE	SEVERITY	STANDARDS	DESCRIPTION
1	Binary makes use of insecure API(s)	warning	<b>CWE:</b> CWE-676: Use of Potentially Dangerous Function <b>OWASP Top 10:</b> M7: Client Code Quality <b>OWASP MASVS:</b> MSTG-CODE-8	The binary may contain the following insecure API(s) <code>_fopen</code> , <code>_memcpy</code> , <code>_printf</code> , <code>_sscanf</code> , <code>_strlen</code> , <code>_strncpy</code>
2	Binary makes use of the insecure Random function(s)	warning	<b>CWE:</b> CWE-330: Use of Insufficiently Random Values <b>OWASP Top 10:</b> M5: Insufficient Cryptography <b>OWASP MASVS:</b> MSTG-CRYPTO-6	The binary may use the following insecure Random function(s) <code>_random</code>
3	Binary makes use of Logging function	info	<b>CWE:</b> CWE-532: Insertion of Sensitive Information into Log File <b>OWASP MASVS:</b> MSTG-STORAGE-3	The binary may use <code>_NSLog</code> function for logging.
4	Binary makes use of malloc function	warning	<b>CWE:</b> CWE-789: Uncontrolled Memory Allocation <b>OWASP Top 10:</b> M7: Client Code Quality <b>OWASP MASVS:</b> MSTG-CODE-8	The binary may use <code>_malloc</code> function instead of <code>calloc</code>

## IPA BINARY ANALYSIS

PROTECTION	STATUS	SEVERITY	DESCRIPTION
NX	False	info	The binary does not have NX bit set. NX bit offer protection against exploitation of memory corruption vulnerabilities by marking memory page as non-executable. However iOS never allows an app to execute from writeable memory. You do not need to specifically enable the 'NX bit' because it's always enabled for all third-party code.
PIE	True	info	The binary is build with <code>-fPIC</code> flag which enables Position independent code. This makes Return Oriented Programming (ROP) attacks much more difficult to execute reliably.
STACK CANARY	True	info	This binary has a stack canary value added to the stack so that it will be overwritten by a stack buffer that overflows the return address. This allows detection of overflows by verifying the integrity of the canary before function return.
ARC	True	info	The binary is compiled with Automatic Reference Counting (ARC) flag. ARC is a compiler feature that provides automatic memory management of Objective-C objects and is an exploit mitigation mechanism against memory corruption vulnerabilities.
RPATH	True	warning	The binary has Runpath Search Path ( <code>@rpath</code> ) set. In certain cases an attacker can abuse this feature to run arbitrary executable for code execution and privilege escalation. Remove the compiler option <code>-rpath</code> to remove <code>@rpath</code> .
CODE SIGNATURE	True	info	This binary has a code signature.
ENCRYPTED	True	info	This binary is encrypted.
SYMBOLS STRIPPED	False	warning	Debug Symbols are available. To strip debugging symbols, set Strip Debug Symbols During Copy to YES, Deployment Postprocessing to YES, and Strip Linked Product to YES in project's build settings.

## 🚩 DYNAMIC LIBRARY & FRAMEWORK BINARY ANALYSIS

NO	DYLIB/Framework	NX	STACK CANARY	ARC	RPATH	CODE SIGNATURE	ENCRYPTED	SYMBOLS STRIPPED
1	Payload/S4Cleaner.app/Frameworks/Kingfisher.framework/Kingfisher	<p>False <a href="#">info</a></p> <p>The binary does not have NX bit set. NX bit offer protection against exploitation of memory corruption vulnerabilities by marking memory page as non-executable. However iOS never allows an app to execute from writeable memory. You do not need to specifically enable the 'NX bit' because it's always enabled for all third-party code.</p>	<p>True <a href="#">info</a></p> <p>This binary has a stack canary value added to the stack so that it will be overwritten by a stack buffer that overflows the return address. This allows detection of overflows by verifying the integrity of the canary before function return.</p>	<p>True <a href="#">info</a></p> <p>The binary is compiled with Automatic Reference Counting (ARC) flag. ARC is a compiler feature that provides automatic memory management of Objective-C objects and is an exploit mitigation mechanism against memory corruption vulnerabilities.</p>	<p>True <a href="#">warning</a></p> <p>The binary has Runpath Search Path (@rpath) set. In certain cases an attacker can abuse this feature to run arbitrary executable for code execution and privilege escalation. Remove the compiler option -rpath to remove @rpath.</p>	<p>True <a href="#">info</a></p> <p>This binary has a code signature.</p>	<p>True <a href="#">info</a></p> <p>This binary is encrypted.</p>	<p>False <a href="#">warning</a></p> <p>Debug Symbols are available. To strip debugging symbols, set Strip Debug Symbols During Copy to YES, Deployment Postprocessing to YES, and Strip Linked Product to YES in project's build settings.</p>

NO	DYLIB/Framework	NX	STACK CANARY	ARC	RPATH	CODE SIGNATURE	ENCRYPTED	SYMBOLS STRIPPED
2	Payload/S4Cleaner.app/Frameworks/GoogleDataTransport.framework/GoogleDataTransport	<p>False <a href="#">info</a></p> <p>The binary does not have NX bit set. NX bit offer protection against exploitation of memory corruption vulnerabilities by marking memory page as non-executable. However iOS never allows an app to execute from writeable memory. You do not need to specifically enable the 'NX bit' because it's always enabled for all third-party code.</p>	<p>True <a href="#">info</a></p> <p>This binary has a stack canary value added to the stack so that it will be overwritten by a stack buffer that overflows the return address. This allows detection of overflows by verifying the integrity of the canary before function return.</p>	<p>True <a href="#">info</a></p> <p>The binary is compiled with Automatic Reference Counting (ARC) flag. ARC is a compiler feature that provides automatic memory management of Objective-C objects and is an exploit mitigation mechanism against memory corruption vulnerabilities.</p>	<p>True <a href="#">warning</a></p> <p>The binary has Runpath Search Path (@rpath) set. In certain cases an attacker can abuse this feature to run arbitrary executable for code execution and privilege escalation. Remove the compiler option -rpath to remove @rpath.</p>	<p>True <a href="#">info</a></p> <p>This binary has a code signature.</p>	<p>True <a href="#">info</a></p> <p>This binary is encrypted.</p>	<p>False <a href="#">warning</a></p> <p>Debug Symbols are available. To strip debugging symbols, set Strip Debug Symbols During Copy to YES, Deployment Postprocessing to YES, and Strip Linked Product to YES in project's build settings.</p>

NO	DYLIB/Framework	NX	STACK CANARY	ARC	RPATH	CODE SIGNATURE	ENCRYPTED	SYMBOLS STRIPPED
3	Payload/S4Cleaner.app/Frameworks/GCDWebServer.framework/GCDWebServer	<p>False <a href="#">info</a></p> <p>The binary does not have NX bit set. NX bit offer protection against exploitation of memory corruption vulnerabilities by marking memory page as non-executable. However iOS never allows an app to execute from writeable memory. You do not need to specifically enable the 'NX bit' because it's always enabled for all third-party code.</p>	<p>True <a href="#">info</a></p> <p>This binary has a stack canary value added to the stack so that it will be overwritten by a stack buffer that overflows the return address. This allows detection of overflows by verifying the integrity of the canary before function return.</p>	<p>True <a href="#">info</a></p> <p>The binary is compiled with Automatic Reference Counting (ARC) flag. ARC is a compiler feature that provides automatic memory management of Objective-C objects and is an exploit mitigation mechanism against memory corruption vulnerabilities.</p>	<p>True <a href="#">warning</a></p> <p>The binary has Runpath Search Path (@rpath) set. In certain cases an attacker can abuse this feature to run arbitrary executable for code execution and privilege escalation. Remove the compiler option -rpath to remove @rpath.</p>	<p>True <a href="#">info</a></p> <p>This binary has a code signature.</p>	<p>True <a href="#">info</a></p> <p>This binary is encrypted.</p>	<p>False <a href="#">warning</a></p> <p>Debug Symbols are available. To strip debugging symbols, set Strip Debug Symbols During Copy to YES, Deployment Postprocessing to YES, and Strip Linked Product to YES in project's build settings.</p>

NO	DYLIB/Framework	NX	STACK CANARY	ARC	RPATH	CODE SIGNATURE	ENCRYPTED	SYMBOLS STRIPPED
4	Payload/S4Cleaner.app/Frameworks/Alamofire.framework/Alamofire	<p>False <a href="#">info</a></p> <p>The binary does not have NX bit set. NX bit offer protection against exploitation of memory corruption vulnerabilities by marking memory page as non-executable. However iOS never allows an app to execute from writeable memory. You do not need to specifically enable the 'NX bit' because it's always enabled for all third-party code.</p>	<p>True <a href="#">info</a></p> <p>This binary has a stack canary value added to the stack so that it will be overwritten by a stack buffer that overflows the return address. This allows detection of overflows by verifying the integrity of the canary before function return.</p>	<p>True <a href="#">info</a></p> <p>The binary is compiled with Automatic Reference Counting (ARC) flag. ARC is a compiler feature that provides automatic memory management of Objective-C objects and is an exploit mitigation mechanism against memory corruption vulnerabilities.</p>	<p>True <a href="#">warning</a></p> <p>The binary has Runpath Search Path (@rpath) set. In certain cases an attacker can abuse this feature to run arbitrary executable for code execution and privilege escalation. Remove the compiler option -rpath to remove @rpath.</p>	<p>True <a href="#">info</a></p> <p>This binary has a code signature.</p>	<p>True <a href="#">info</a></p> <p>This binary is encrypted.</p>	<p>False <a href="#">warning</a></p> <p>Debug Symbols are available. To strip debugging symbols, set Strip Debug Symbols During Copy to YES, Deployment Postprocessing to YES, and Strip Linked Product to YES in project's build settings.</p>



NO	DYLIB/Framework	NX	STACK CANARY	ARC	RPATH	CODE SIGNATURE	ENCRYPTED	SYMBOLS STRIPPED
5	Payload/S4Cleaner.app/Frameworks/FBAEMKit.framework/FBAEMKit	<p>False <a href="#">info</a></p> <p>The binary does not have NX bit set. NX bit offer protection against exploitation of memory corruption vulnerabilities by marking memory page as non-executable. However iOS never allows an app to execute from writeable memory. You do not need to specifically enable the 'NX bit' because it's always enabled for all third-party code.</p>	<p>True <a href="#">info</a></p> <p>This binary has a stack canary value added to the stack so that it will be overwritten by a stack buffer that overflows the return address. This allows detection of overflows by verifying the integrity of the canary before function return.</p>	<p>True <a href="#">info</a></p> <p>The binary is compiled with Automatic Reference Counting (ARC) flag. ARC is a compiler feature that provides automatic memory management of Objective-C objects and is an exploit mitigation mechanism against memory corruption vulnerabilities.</p>	<p>True <a href="#">warning</a></p> <p>The binary has Runpath Search Path (@rpath) set. In certain cases an attacker can abuse this feature to run arbitrary executable for code execution and privilege escalation. Remove the compiler option -rpath to remove @rpath.</p>	<p>True <a href="#">info</a></p> <p>This binary has a code signature.</p>	<p>True <a href="#">info</a></p> <p>This binary is encrypted.</p>	<p>False <a href="#">warning</a></p> <p>Debug Symbols are available. To strip debugging symbols, set Strip Debug Symbols During Copy to YES, Deployment Postprocessing to YES, and Strip Linked Product to YES in project's build settings.</p>

NO	DYLIB/Framework	NX	STACK CANARY	ARC	RPATH	CODE SIGNATURE	ENCRYPTED	SYMBOLS STRIPPED
6	Payload/S4Cleaner.app/Frameworks/FirebaseSessions.framework/FirebaseSessions	<p>False <a href="#">info</a></p> <p>The binary does not have NX bit set. NX bit offer protection against exploitation of memory corruption vulnerabilities by marking memory page as non-executable. However iOS never allows an app to execute from writeable memory. You do not need to specifically enable the 'NX bit' because it's always enabled for all third-party code.</p>	<p>True <a href="#">info</a></p> <p>This binary has a stack canary value added to the stack so that it will be overwritten by a stack buffer that overflows the return address. This allows detection of overflows by verifying the integrity of the canary before function return.</p>	<p>True <a href="#">info</a></p> <p>The binary is compiled with Automatic Reference Counting (ARC) flag. ARC is a compiler feature that provides automatic memory management of Objective-C objects and is an exploit mitigation mechanism against memory corruption vulnerabilities.</p>	<p>True <a href="#">warning</a></p> <p>The binary has Runpath Search Path (@rpath) set. In certain cases an attacker can abuse this feature to run arbitrary executable for code execution and privilege escalation. Remove the compiler option -rpath to remove @rpath.</p>	<p>True <a href="#">info</a></p> <p>This binary has a code signature.</p>	<p>True <a href="#">info</a></p> <p>This binary is encrypted.</p>	<p>False <a href="#">warning</a></p> <p>Debug Symbols are available. To strip debugging symbols, set Strip Debug Symbols During Copy to YES, Deployment Postprocessing to YES, and Strip Linked Product to YES in project's build settings.</p>

NO	DYLIB/Framework	NX	STACK CANARY	ARC	RPATH	CODE SIGNATURE	ENCRYPTED	SYMBOLS STRIPPED
7	Payload/S4Cleaner.app/Frameworks/SwiftUIIntrospect.framework/SwiftUIIntrospect	<p>False <a href="#">info</a></p> <p>The binary does not have NX bit set. NX bit offer protection against exploitation of memory corruption vulnerabilities by marking memory page as non-executable. However iOS never allows an app to execute from writeable memory. You do not need to specifically enable the 'NX bit' because it's always enabled for all third-party code.</p>	<p>True <a href="#">info</a></p> <p>This binary has a stack canary value added to the stack so that it will be overwritten by a stack buffer that overflows the return address. This allows detection of overflows by verifying the integrity of the canary before function return.</p>	<p>True <a href="#">info</a></p> <p>The binary is compiled with Automatic Reference Counting (ARC) flag. ARC is a compiler feature that provides automatic memory management of Objective-C objects and is an exploit mitigation mechanism against memory corruption vulnerabilities.</p>	<p>True <a href="#">warning</a></p> <p>The binary has Runpath Search Path (@rpath) set. In certain cases an attacker can abuse this feature to run arbitrary executable for code execution and privilege escalation. Remove the compiler option -rpath to remove @rpath.</p>	<p>True <a href="#">info</a></p> <p>This binary has a code signature.</p>	<p>True <a href="#">info</a></p> <p>This binary is encrypted.</p>	<p>False <a href="#">warning</a></p> <p>Debug Symbols are available. To strip debugging symbols, set Strip Debug Symbols During Copy to YES, Deployment Postprocessing to YES, and Strip Linked Product to YES in project's build settings.</p>

NO	DYLIB/Framework	NX	STACK CANARY	ARC	RPATH	CODE SIGNATURE	ENCRYPTED	SYMBOLS STRIPPED
8	Payload/S4Cleaner.app/Frameworks/KeychainAccess.framework/KeychainAccess	<p>False <a href="#">info</a></p> <p>The binary does not have NX bit set. NX bit offer protection against exploitation of memory corruption vulnerabilities by marking memory page as non-executable. However iOS never allows an app to execute from writeable memory. You do not need to specifically enable the 'NX bit' because it's always enabled for all third-party code.</p>	<p>True <a href="#">info</a></p> <p>This binary has a stack canary value added to the stack so that it will be overwritten by a stack buffer that overflows the return address. This allows detection of overflows by verifying the integrity of the canary before function return.</p>	<p>True <a href="#">info</a></p> <p>The binary is compiled with Automatic Reference Counting (ARC) flag. ARC is a compiler feature that provides automatic memory management of Objective-C objects and is an exploit mitigation mechanism against memory corruption vulnerabilities.</p>	<p>True <a href="#">warning</a></p> <p>The binary has Runpath Search Path (@rpath) set. In certain cases an attacker can abuse this feature to run arbitrary executable for code execution and privilege escalation. Remove the compiler option -rpath to remove @rpath.</p>	<p>True <a href="#">info</a></p> <p>This binary has a code signature.</p>	<p>True <a href="#">info</a></p> <p>This binary is encrypted.</p>	<p>False <a href="#">warning</a></p> <p>Debug Symbols are available. To strip debugging symbols, set Strip Debug Symbols During Copy to YES, Deployment Postprocessing to YES, and Strip Linked Product to YES in project's build settings.</p>

NO	DYLIB/Framework	NX	STACK CANARY	ARC	RPATH	CODE SIGNATURE	ENCRYPTED	SYMBOLS STRIPPED
9	Payload/S4Cleaner.app/Frameworks/nanopb.framework/nanopb	<p>False <a href="#">info</a></p> <p>The binary does not have NX bit set. NX bit offer protection against exploitation of memory corruption vulnerabilities by marking memory page as non-executable. However iOS never allows an app to execute from writeable memory. You do not need to specifically enable the 'NX bit' because it's always enabled for all third-party code.</p>	<p>True <a href="#">info</a></p> <p>This binary has a stack canary value added to the stack so that it will be overwritten by a stack buffer that overflows the return address. This allows detection of overflows by verifying the integrity of the canary before function return.</p>	<p>False <b>high</b></p> <p>The binary is not compiled with Automatic Reference Counting (ARC) flag. ARC is a compiler feature that provides automatic memory management of Objective-C objects and protects from memory corruption vulnerabilities. Use compiler option -fobjc-arc to enable ARC or set Objective-C Automatic Reference Counting to YES in project configuration.</p>	<p>True <b>warning</b></p> <p>The binary has Runpath Search Path (@rpath) set. In certain cases an attacker can abuse this feature to run arbitrary executable for code execution and privilege escalation. Remove the compiler option -rpath to remove @rpath.</p>	<p>True <a href="#">info</a></p> <p>This binary has a code signature.</p>	<p>True <a href="#">info</a></p> <p>This binary is encrypted.</p>	<p>False <b>warning</b></p> <p>Debug Symbols are available. To strip debugging symbols, set Strip Debug Symbols During Copy to YES, Deployment Postprocessing to YES, and Strip Linked Product to YES in project's build settings.</p>

NO	DYLIB/Framework	NX	STACK CANARY	ARC	RPATH	CODE SIGNATURE	ENCRYPTED	SYMBOLS STRIPPED
10	Payload/S4Cleaner.app/Frameworks/NevSwipeBackGesture.framework/NevSwipeBackGesture	<p>False <a href="#">info</a></p> <p>The binary does not have NX bit set. NX bit offer protection against exploitation of memory corruption vulnerabilities by marking memory page as non-executable. However iOS never allows an app to execute from writeable memory. You do not need to specifically enable the 'NX bit' because it's always enabled for all third-party code.</p>	<p>False <b>high</b></p> <p>This binary does not have a stack canary value added to the stack. Stack canaries are used to detect and prevent exploits from overwriting return address. Use the option -fstack-protector-all to enable stack canaries.</p>	<p>True <a href="#">info</a></p> <p>The binary is compiled with Automatic Reference Counting (ARC) flag. ARC is a compiler feature that provides automatic memory management of Objective-C objects and is an exploit mitigation mechanism against memory corruption vulnerabilities.</p>	<p>True <b>warning</b></p> <p>The binary has Runpath Search Path (@rpath) set. In certain cases an attacker can abuse this feature to run arbitrary executable for code execution and privilege escalation. Remove the compiler option -rpath to remove @rpath.</p>	<p>True <a href="#">info</a></p> <p>This binary has a code signature.</p>	<p>True <a href="#">info</a></p> <p>This binary is encrypted.</p>	<p>False <b>warning</b></p> <p>Debug Symbols are available. To strip debugging symbols, set Strip Debug Symbols During Copy to YES, Deployment Postprocessing to YES, and Strip Linked Product to YES in project's build settings.</p>

NO	DYLIB/Framework	NX	STACK CANARY	ARC	RPATH	CODE SIGNATURE	ENCRYPTED	SYMBOLS STRIPPED
11	Payload/S4Cleaner.app/Frameworks/CollectionViewPagingLayout.framework/CollectionViewPagingLayout	<p>False <a href="#">info</a></p> <p>The binary does not have NX bit set. NX bit offer protection against exploitation of memory corruption vulnerabilities by marking memory page as non-executable. However iOS never allows an app to execute from writeable memory. You do not need to specifically enable the 'NX bit' because it's always enabled for all third-party code.</p>	<p>True <a href="#">info</a></p> <p>This binary has a stack canary value added to the stack so that it will be overwritten by a stack buffer that overflows the return address. This allows detection of overflows by verifying the integrity of the canary before function return.</p>	<p>True <a href="#">info</a></p> <p>The binary is compiled with Automatic Reference Counting (ARC) flag. ARC is a compiler feature that provides automatic memory management of Objective-C objects and is an exploit mitigation mechanism against memory corruption vulnerabilities.</p>	<p>True <a href="#">warning</a></p> <p>The binary has Runpath Search Path (@rpath) set. In certain cases an attacker can abuse this feature to run arbitrary executable for code execution and privilege escalation. Remove the compiler option -rpath to remove @rpath.</p>	<p>True <a href="#">info</a></p> <p>This binary has a code signature.</p>	<p>True <a href="#">info</a></p> <p>This binary is encrypted.</p>	<p>False <a href="#">warning</a></p> <p>Debug Symbols are available. To strip debugging symbols, set Strip Debug Symbols During Copy to YES, Deployment Postprocessing to YES, and Strip Linked Product to YES in project's build settings.</p>

NO	DYLIB/Framework	NX	STACK CANARY	ARC	RPATH	CODE SIGNATURE	ENCRYPTED	SYMBOLS STRIPPED
12	Payload/S4Cleaner.app/Frameworks/CardStack.framework/CardStack	<p>False <a href="#">info</a></p> <p>The binary does not have NX bit set. NX bit offer protection against exploitation of memory corruption vulnerabilities by marking memory page as non-executable. However iOS never allows an app to execute from writeable memory. You do not need to specifically enable the 'NX bit' because it's always enabled for all third-party code.</p>	<p>False <a href="#">high</a></p> <p>This binary does not have a stack canary value added to the stack. Stack canaries are used to detect and prevent exploits from overwriting return address. Use the option -fstack-protector-all to enable stack canaries.</p>	<p>True <a href="#">info</a></p> <p>The binary is compiled with Automatic Reference Counting (ARC) flag. ARC is a compiler feature that provides automatic memory management of Objective-C objects and is an exploit mitigation mechanism against memory corruption vulnerabilities.</p>	<p>True <a href="#">warning</a></p> <p>The binary has Runpath Search Path (@rpath) set. In certain cases an attacker can abuse this feature to run arbitrary executable for code execution and privilege escalation. Remove the compiler option -rpath to remove @rpath.</p>	<p>True <a href="#">info</a></p> <p>This binary has a code signature.</p>	<p>True <a href="#">info</a></p> <p>This binary is encrypted.</p>	<p>False <a href="#">warning</a></p> <p>Debug Symbols are available. To strip debugging symbols, set Strip Debug Symbols During Copy to YES, Deployment Postprocessing to YES, and Strip Linked Product to YES in project's build settings.</p>



NO	DYLIB/Framework	NX	STACK CANARY	ARC	RPATH	CODE SIGNATURE	ENCRYPTED	SYMBOLS STRIPPED
13	Payload/S4Cleaner.app/Frameworks/Lottie.framework/Lottie	<p>False <a href="#">info</a></p> <p>The binary does not have NX bit set. NX bit offer protection against exploitation of memory corruption vulnerabilities by marking memory page as non-executable. However iOS never allows an app to execute from writeable memory. You do not need to specifically enable the 'NX bit' because it's always enabled for all third-party code.</p>	<p>True <a href="#">info</a></p> <p>This binary has a stack canary value added to the stack so that it will be overwritten by a stack buffer that overflows the return address. This allows detection of overflows by verifying the integrity of the canary before function return.</p>	<p>True <a href="#">info</a></p> <p>The binary is compiled with Automatic Reference Counting (ARC) flag. ARC is a compiler feature that provides automatic memory management of Objective-C objects and is an exploit mitigation mechanism against memory corruption vulnerabilities.</p>	<p>True <a href="#">warning</a></p> <p>The binary has Runpath Search Path (@rpath) set. In certain cases an attacker can abuse this feature to run arbitrary executable for code execution and privilege escalation. Remove the compiler option -rpath to remove @rpath.</p>	<p>True <a href="#">info</a></p> <p>This binary has a code signature.</p>	<p>True <a href="#">info</a></p> <p>This binary is encrypted.</p>	<p>False <a href="#">warning</a></p> <p>Debug Symbols are available. To strip debugging symbols, set Strip Debug Symbols During Copy to YES, Deployment Postprocessing to YES, and Strip Linked Product to YES in project's build settings.</p>

NO	DYLIB/Framework	NX	STACK CANARY	ARC	RPATH	CODE SIGNATURE	ENCRYPTED	SYMBOLS STRIPPED
14	Payload/S4Cleaner.app/Frameworks/Promises.framework/Promises	<p>False <a href="#">info</a></p> <p>The binary does not have NX bit set. NX bit offer protection against exploitation of memory corruption vulnerabilities by marking memory page as non-executable. However iOS never allows an app to execute from writeable memory. You do not need to specifically enable the 'NX bit' because it's always enabled for all third-party code.</p>	<p>True <a href="#">info</a></p> <p>This binary has a stack canary value added to the stack so that it will be overwritten by a stack buffer that overflows the return address. This allows detection of overflows by verifying the integrity of the canary before function return.</p>	<p>True <a href="#">info</a></p> <p>The binary is compiled with Automatic Reference Counting (ARC) flag. ARC is a compiler feature that provides automatic memory management of Objective-C objects and is an exploit mitigation mechanism against memory corruption vulnerabilities.</p>	<p>True <a href="#">warning</a></p> <p>The binary has Runpath Search Path (@rpath) set. In certain cases an attacker can abuse this feature to run arbitrary executable for code execution and privilege escalation. Remove the compiler option -rpath to remove @rpath.</p>	<p>True <a href="#">info</a></p> <p>This binary has a code signature.</p>	<p>True <a href="#">info</a></p> <p>This binary is encrypted.</p>	<p>False <a href="#">warning</a></p> <p>Debug Symbols are available. To strip debugging symbols, set Strip Debug Symbols During Copy to YES, Deployment Postprocessing to YES, and Strip Linked Product to YES in project's build settings.</p>

NO	DYLIB/Framework	NX	STACK CANARY	ARC	RPATH	CODE SIGNATURE	ENCRYPTED	SYMBOLS STRIPPED
15	Payload/S4Cleaner.app/Frameworks/FirebaseABTesting.framework/FirebaseABTesting	<p>False <a href="#">info</a></p> <p>The binary does not have NX bit set. NX bit offer protection against exploitation of memory corruption vulnerabilities by marking memory page as non-executable. However iOS never allows an app to execute from writeable memory. You do not need to specifically enable the 'NX bit' because it's always enabled for all third-party code.</p>	<p>True <a href="#">info</a></p> <p>This binary has a stack canary value added to the stack so that it will be overwritten by a stack buffer that overflows the return address. This allows detection of overflows by verifying the integrity of the canary before function return.</p>	<p>True <a href="#">info</a></p> <p>The binary is compiled with Automatic Reference Counting (ARC) flag. ARC is a compiler feature that provides automatic memory management of Objective-C objects and is an exploit mitigation mechanism against memory corruption vulnerabilities.</p>	<p>True <a href="#">warning</a></p> <p>The binary has Runpath Search Path (@rpath) set. In certain cases an attacker can abuse this feature to run arbitrary executable for code execution and privilege escalation. Remove the compiler option -rpath to remove @rpath.</p>	<p>True <a href="#">info</a></p> <p>This binary has a code signature.</p>	<p>True <a href="#">info</a></p> <p>This binary is encrypted.</p>	<p>False <a href="#">warning</a></p> <p>Debug Symbols are available. To strip debugging symbols, set Strip Debug Symbols During Copy to YES, Deployment Postprocessing to YES, and Strip Linked Product to YES in project's build settings.</p>

NO	DYLIB/Framework	NX	STACK CANARY	ARC	RPATH	CODE SIGNATURE	ENCRYPTED	SYMBOLS STRIPPED
16	Payload/S4Cleaner.app/Frameworks/ExytePopupView.framework/ExytePopupView	<p>False <a href="#">info</a></p> <p>The binary does not have NX bit set. NX bit offer protection against exploitation of memory corruption vulnerabilities by marking memory page as non-executable. However iOS never allows an app to execute from writeable memory. You do not need to specifically enable the 'NX bit' because it's always enabled for all third-party code.</p>	<p>True <a href="#">info</a></p> <p>This binary has a stack canary value added to the stack so that it will be overwritten by a stack buffer that overflows the return address. This allows detection of overflows by verifying the integrity of the canary before function return.</p>	<p>True <a href="#">info</a></p> <p>The binary is compiled with Automatic Reference Counting (ARC) flag. ARC is a compiler feature that provides automatic memory management of Objective-C objects and is an exploit mitigation mechanism against memory corruption vulnerabilities.</p>	<p>True <a href="#">warning</a></p> <p>The binary has Runpath Search Path (@rpath) set. In certain cases an attacker can abuse this feature to run arbitrary executable for code execution and privilege escalation. Remove the compiler option -rpath to remove @rpath.</p>	<p>True <a href="#">info</a></p> <p>This binary has a code signature.</p>	<p>True <a href="#">info</a></p> <p>This binary is encrypted.</p>	<p>False <a href="#">warning</a></p> <p>Debug Symbols are available. To strip debugging symbols, set Strip Debug Symbols During Copy to YES, Deployment Postprocessing to YES, and Strip Linked Product to YES in project's build settings.</p>

NO	DYLIB/Framework	NX	STACK CANARY	ARC	RPATH	CODE SIGNATURE	ENCRYPTED	SYMBOLS STRIPPED
17	Payload/S4Cleaner.app/Frameworks/JKCategories.framework/JKCategories	<p>False <a href="#">info</a></p> <p>The binary does not have NX bit set. NX bit offer protection against exploitation of memory corruption vulnerabilities by marking memory page as non-executable. However iOS never allows an app to execute from writeable memory. You do not need to specifically enable the 'NX bit' because it's always enabled for all third-party code.</p>	<p>True <a href="#">info</a></p> <p>This binary has a stack canary value added to the stack so that it will be overwritten by a stack buffer that overflows the return address. This allows detection of overflows by verifying the integrity of the canary before function return.</p>	<p>True <a href="#">info</a></p> <p>The binary is compiled with Automatic Reference Counting (ARC) flag. ARC is a compiler feature that provides automatic memory management of Objective-C objects and is an exploit mitigation mechanism against memory corruption vulnerabilities.</p>	<p>True <a href="#">warning</a></p> <p>The binary has Runpath Search Path (@rpath) set. In certain cases an attacker can abuse this feature to run arbitrary executable for code execution and privilege escalation. Remove the compiler option -rpath to remove @rpath.</p>	<p>True <a href="#">info</a></p> <p>This binary has a code signature.</p>	<p>True <a href="#">info</a></p> <p>This binary is encrypted.</p>	<p>False <a href="#">warning</a></p> <p>Debug Symbols are available. To strip debugging symbols, set Strip Debug Symbols During Copy to YES, Deployment Postprocessing to YES, and Strip Linked Product to YES in project's build settings.</p>

NO	DYLIB/Framework	NX	STACK CANARY	ARC	RPATH	CODE SIGNATURE	ENCRYPTED	SYMBOLS STRIPPED
18	Payload/S4Cleaner.app/Frameworks/FirebaseSharedSwift.framework/FirebaseSharedSwift	<p>False <a href="#">info</a></p> <p>The binary does not have NX bit set. NX bit offer protection against exploitation of memory corruption vulnerabilities by marking memory page as non-executable. However iOS never allows an app to execute from writeable memory. You do not need to specifically enable the 'NX bit' because it's always enabled for all third-party code.</p>	<p>True <a href="#">info</a></p> <p>This binary has a stack canary value added to the stack so that it will be overwritten by a stack buffer that overflows the return address. This allows detection of overflows by verifying the integrity of the canary before function return.</p>	<p>True <a href="#">info</a></p> <p>The binary is compiled with Automatic Reference Counting (ARC) flag. ARC is a compiler feature that provides automatic memory management of Objective-C objects and is an exploit mitigation mechanism against memory corruption vulnerabilities.</p>	<p>True <a href="#">warning</a></p> <p>The binary has Runpath Search Path (@rpath) set. In certain cases an attacker can abuse this feature to run arbitrary executable for code execution and privilege escalation. Remove the compiler option -rpath to remove @rpath.</p>	<p>True <a href="#">info</a></p> <p>This binary has a code signature.</p>	<p>True <a href="#">info</a></p> <p>This binary is encrypted.</p>	<p>False <a href="#">warning</a></p> <p>Debug Symbols are available. To strip debugging symbols, set Strip Debug Symbols During Copy to YES, Deployment Postprocessing to YES, and Strip Linked Product to YES in project's build settings.</p>

NO	DYLIB/Framework	NX	STACK CANARY	ARC	RPATH	CODE SIGNATURE	ENCRYPTED	SYMBOLS STRIPPED
19	Payload/S4Cleaner.app/Frameworks/FirebaseCoreInternal.framework/FirebaseCoreInternal	<p>False <a href="#">info</a></p> <p>The binary does not have NX bit set. NX bit offer protection against exploitation of memory corruption vulnerabilities by marking memory page as non-executable. However iOS never allows an app to execute from writeable memory. You do not need to specifically enable the 'NX bit' because it's always enabled for all third-party code.</p>	<p>True <a href="#">info</a></p> <p>This binary has a stack canary value added to the stack so that it will be overwritten by a stack buffer that overflows the return address. This allows detection of overflows by verifying the integrity of the canary before function return.</p>	<p>True <a href="#">info</a></p> <p>The binary is compiled with Automatic Reference Counting (ARC) flag. ARC is a compiler feature that provides automatic memory management of Objective-C objects and is an exploit mitigation mechanism against memory corruption vulnerabilities.</p>	<p>True <a href="#">warning</a></p> <p>The binary has Runpath Search Path (@rpath) set. In certain cases an attacker can abuse this feature to run arbitrary executable for code execution and privilege escalation. Remove the compiler option -rpath to remove @rpath.</p>	<p>True <a href="#">info</a></p> <p>This binary has a code signature.</p>	<p>True <a href="#">info</a></p> <p>This binary is encrypted.</p>	<p>False <a href="#">warning</a></p> <p>Debug Symbols are available. To strip debugging symbols, set Strip Debug Symbols During Copy to YES, Deployment Postprocessing to YES, and Strip Linked Product to YES in project's build settings.</p>

NO	DYLIB/Framework	NX	STACK CANARY	ARC	RPATH	CODE SIGNATURE	ENCRYPTED	SYMBOLS STRIPPED
20	Payload/S4Cleaner.app/Frameworks/FirebaseRemoteConfig.framework/FirebaseRemoteConfig	<p>False <a href="#">info</a></p> <p>The binary does not have NX bit set. NX bit offer protection against exploitation of memory corruption vulnerabilities by marking memory page as non-executable. However iOS never allows an app to execute from writeable memory. You do not need to specifically enable the 'NX bit' because it's always enabled for all third-party code.</p>	<p>True <a href="#">info</a></p> <p>This binary has a stack canary value added to the stack so that it will be overwritten by a stack buffer that overflows the return address. This allows detection of overflows by verifying the integrity of the canary before function return.</p>	<p>True <a href="#">info</a></p> <p>The binary is compiled with Automatic Reference Counting (ARC) flag. ARC is a compiler feature that provides automatic memory management of Objective-C objects and is an exploit mitigation mechanism against memory corruption vulnerabilities.</p>	<p>True <a href="#">warning</a></p> <p>The binary has Runpath Search Path (@rpath) set. In certain cases an attacker can abuse this feature to run arbitrary executable for code execution and privilege escalation. Remove the compiler option -rpath to remove @rpath.</p>	<p>True <a href="#">info</a></p> <p>This binary has a code signature.</p>	<p>True <a href="#">info</a></p> <p>This binary is encrypted.</p>	<p>False <a href="#">warning</a></p> <p>Debug Symbols are available. To strip debugging symbols, set Strip Debug Symbols During Copy to YES, Deployment Postprocessing to YES, and Strip Linked Product to YES in project's build settings.</p>



NO	DYLIB/Framework	NX	STACK CANARY	ARC	RPATH	CODE SIGNATURE	ENCRYPTED	SYMBOLS STRIPPED
21	Payload/S4Cleaner.app/Frameworks/SLImageCompare.framework/SLImageCompare	<p>False <a href="#">info</a></p> <p>The binary does not have NX bit set. NX bit offer protection against exploitation of memory corruption vulnerabilities by marking memory page as non-executable. However iOS never allows an app to execute from writeable memory. You do not need to specifically enable the 'NX bit' because it's always enabled for all third-party code.</p>	<p>True <a href="#">info</a></p> <p>This binary has a stack canary value added to the stack so that it will be overwritten by a stack buffer that overflows the return address. This allows detection of overflows by verifying the integrity of the canary before function return.</p>	<p>True <a href="#">info</a></p> <p>The binary is compiled with Automatic Reference Counting (ARC) flag. ARC is a compiler feature that provides automatic memory management of Objective-C objects and is an exploit mitigation mechanism against memory corruption vulnerabilities.</p>	<p>True <a href="#">warning</a></p> <p>The binary has Runpath Search Path (@rpath) set. In certain cases an attacker can abuse this feature to run arbitrary executable for code execution and privilege escalation. Remove the compiler option -rpath to remove @rpath.</p>	<p>True <a href="#">info</a></p> <p>This binary has a code signature.</p>	<p>True <a href="#">info</a></p> <p>This binary is encrypted.</p>	<p>False <a href="#">warning</a></p> <p>Debug Symbols are available. To strip debugging symbols, set Strip Debug Symbols During Copy to YES, Deployment Postprocessing to YES, and Strip Linked Product to YES in project's build settings.</p>

NO	DYLIB/Framework	NX	STACK CANARY	ARC	RPATH	CODE SIGNATURE	ENCRYPTED	SYMBOLS STRIPPED
22	Payload/S4Cleaner.app/Frameworks/FBLPromises.framework/FBLPromises	<p>False <a href="#">info</a></p> <p>The binary does not have NX bit set. NX bit offer protection against exploitation of memory corruption vulnerabilities by marking memory page as non-executable. However iOS never allows an app to execute from writeable memory. You do not need to specifically enable the 'NX bit' because it's always enabled for all third-party code.</p>	<p>True <a href="#">info</a></p> <p>This binary has a stack canary value added to the stack so that it will be overwritten by a stack buffer that overflows the return address. This allows detection of overflows by verifying the integrity of the canary before function return.</p>	<p>True <a href="#">info</a></p> <p>The binary is compiled with Automatic Reference Counting (ARC) flag. ARC is a compiler feature that provides automatic memory management of Objective-C objects and is an exploit mitigation mechanism against memory corruption vulnerabilities.</p>	<p>True <a href="#">warning</a></p> <p>The binary has Runpath Search Path (@rpath) set. In certain cases an attacker can abuse this feature to run arbitrary executable for code execution and privilege escalation. Remove the compiler option -rpath to remove @rpath.</p>	<p>True <a href="#">info</a></p> <p>This binary has a code signature.</p>	<p>True <a href="#">info</a></p> <p>This binary is encrypted.</p>	<p>False <a href="#">warning</a></p> <p>Debug Symbols are available. To strip debugging symbols, set Strip Debug Symbols During Copy to YES, Deployment Postprocessing to YES, and Strip Linked Product to YES in project's build settings.</p>

NO	DYLIB/Framework	NX	STACK CANARY	ARC	RPATH	CODE SIGNATURE	ENCRYPTED	SYMBOLS STRIPPED
23	Payload/S4Cleaner.app/Frameworks/FirebaseCrashlytics.framework/FirebaseCrashlytics	<p>False <a href="#">info</a></p> <p>The binary does not have NX bit set. NX bit offer protection against exploitation of memory corruption vulnerabilities by marking memory page as non-executable. However iOS never allows an app to execute from writeable memory. You do not need to specifically enable the 'NX bit' because it's always enabled for all third-party code.</p>	<p>True <a href="#">info</a></p> <p>This binary has a stack canary value added to the stack so that it will be overwritten by a stack buffer that overflows the return address. This allows detection of overflows by verifying the integrity of the canary before function return.</p>	<p>True <a href="#">info</a></p> <p>The binary is compiled with Automatic Reference Counting (ARC) flag. ARC is a compiler feature that provides automatic memory management of Objective-C objects and is an exploit mitigation mechanism against memory corruption vulnerabilities.</p>	<p>True <a href="#">warning</a></p> <p>The binary has Runpath Search Path (@rpath) set. In certain cases an attacker can abuse this feature to run arbitrary executable for code execution and privilege escalation. Remove the compiler option -rpath to remove @rpath.</p>	<p>True <a href="#">info</a></p> <p>This binary has a code signature.</p>	<p>True <a href="#">info</a></p> <p>This binary is encrypted.</p>	<p>False <a href="#">warning</a></p> <p>Debug Symbols are available. To strip debugging symbols, set Strip Debug Symbols During Copy to YES, Deployment Postprocessing to YES, and Strip Linked Product to YES in project's build settings.</p>

NO	DYLIB/Framework	NX	STACK CANARY	ARC	RPATH	CODE SIGNATURE	ENCRYPTED	SYMBOLS STRIPPED
24	Payload/S4Cleaner.app/Frameworks/FirebaseRemoteConfigInterop.framework/FirebaseRemoteConfigInterop	<p>False <a href="#">info</a></p> <p>The binary does not have NX bit set. NX bit offer protection against exploitation of memory corruption vulnerabilities by marking memory page as non-executable. However iOS never allows an app to execute from writeable memory. You do not need to specifically enable the 'NX bit' because it's always enabled for all third-party code.</p>	<p>False <a href="#">high</a></p> <p>This binary does not have a stack canary value added to the stack. Stack canaries are used to detect and prevent exploits from overwriting return address. Use the option -fstack-protector-all to enable stack canaries.</p>	<p>True <a href="#">info</a></p> <p>The binary is compiled with Automatic Reference Counting (ARC) flag. ARC is a compiler feature that provides automatic memory management of Objective-C objects and is an exploit mitigation mechanism against memory corruption vulnerabilities.</p>	<p>True <a href="#">warning</a></p> <p>The binary has Runpath Search Path (@rpath) set. In certain cases an attacker can abuse this feature to run arbitrary executable for code execution and privilege escalation. Remove the compiler option -rpath to remove @rpath.</p>	<p>True <a href="#">info</a></p> <p>This binary has a code signature.</p>	<p>True <a href="#">info</a></p> <p>This binary is encrypted.</p>	<p>False <a href="#">warning</a></p> <p>Debug Symbols are available. To strip debugging symbols, set Strip Debug Symbols During Copy to YES, Deployment Postprocessing to YES, and Strip Linked Product to YES in project's build settings.</p>

NO	DYLIB/Framework	NX	STACK CANARY	ARC	RPATH	CODE SIGNATURE	ENCRYPTED	SYMBOLS STRIPPED
25	Payload/S4Cleaner.app/Frameworks/AppLovinSDK.framework/AppLovinSDK	<p>False <a href="#">info</a></p> <p>The binary does not have NX bit set. NX bit offer protection against exploitation of memory corruption vulnerabilities by marking memory page as non-executable. However iOS never allows an app to execute from writeable memory. You do not need to specifically enable the 'NX bit' because it's always enabled for all third-party code.</p>	<p>True <a href="#">info</a></p> <p>This binary has a stack canary value added to the stack so that it will be overwritten by a stack buffer that overflows the return address. This allows detection of overflows by verifying the integrity of the canary before function return.</p>	<p>True <a href="#">info</a></p> <p>The binary is compiled with Automatic Reference Counting (ARC) flag. ARC is a compiler feature that provides automatic memory management of Objective-C objects and is an exploit mitigation mechanism against memory corruption vulnerabilities.</p>	<p>True <a href="#">warning</a></p> <p>The binary has Runpath Search Path (@rpath) set. In certain cases an attacker can abuse this feature to run arbitrary executable for code execution and privilege escalation. Remove the compiler option -rpath to remove @rpath.</p>	<p>True <a href="#">info</a></p> <p>This binary has a code signature.</p>	<p>True <a href="#">info</a></p> <p>This binary is encrypted.</p>	<p>False <a href="#">warning</a></p> <p>Debug Symbols are available. To strip debugging symbols, set Strip Debug Symbols During Copy to YES, Deployment Postprocessing to YES, and Strip Linked Product to YES in project's build settings.</p>

NO	DYLIB/Framework	NX	STACK CANARY	ARC	RPATH	CODE SIGNATURE	ENCRYPTED	SYMBOLS STRIPPED
26	Payload/S4Cleaner.app/Frameworks/FirebaseCoreExtension.framework/FirebaseCoreExtension	<p>False <a href="#">info</a></p> <p>The binary does not have NX bit set. NX bit offer protection against exploitation of memory corruption vulnerabilities by marking memory page as non-executable. However iOS never allows an app to execute from writeable memory. You do not need to specifically enable the 'NX bit' because it's always enabled for all third-party code.</p>	<p>False <a href="#">high</a></p> <p>This binary does not have a stack canary value added to the stack. Stack canaries are used to detect and prevent exploits from overwriting return address. Use the option -fstack-protector-all to enable stack canaries.</p>	<p>False <a href="#">high</a></p> <p>The binary is not compiled with Automatic Reference Counting (ARC) flag. ARC is a compiler feature that provides automatic memory management of Objective-C objects and protects from memory corruption vulnerabilities. Use compiler option -fobjc-arc to enable ARC or set Objective-C Automatic Reference Counting to YES in project configuration.</p>	<p>True <a href="#">warning</a></p> <p>The binary has Runpath Search Path (@rpath) set. In certain cases an attacker can abuse this feature to run arbitrary executable for code execution and privilege escalation. Remove the compiler option -rpath to remove @rpath.</p>	<p>True <a href="#">info</a></p> <p>This binary has a code signature.</p>	<p>True <a href="#">info</a></p> <p>This binary is encrypted.</p>	<p>False <a href="#">warning</a></p> <p>Debug Symbols are available. To strip debugging symbols, set Strip Debug Symbols During Copy to YES, Deployment Postprocessing to YES, and Strip Linked Product to YES in project's build settings.</p>

NO	DYLIB/Framework	NX	STACK CANARY	ARC	RPATH	CODE SIGNATURE	ENCRYPTED	SYMBOLS STRIPPED
27	Payload/S4Cleaner.app/Frameworks/SwifterSwift.framework/SwifterSwift	<p>False <a href="#">info</a></p> <p>The binary does not have NX bit set. NX bit offer protection against exploitation of memory corruption vulnerabilities by marking memory page as non-executable. However iOS never allows an app to execute from writeable memory. You do not need to specifically enable the 'NX bit' because it's always enabled for all third-party code.</p>	<p>True <a href="#">info</a></p> <p>This binary has a stack canary value added to the stack so that it will be overwritten by a stack buffer that overflows the return address. This allows detection of overflows by verifying the integrity of the canary before function return.</p>	<p>True <a href="#">info</a></p> <p>The binary is compiled with Automatic Reference Counting (ARC) flag. ARC is a compiler feature that provides automatic memory management of Objective-C objects and is an exploit mitigation mechanism against memory corruption vulnerabilities.</p>	<p>True <a href="#">warning</a></p> <p>The binary has Runpath Search Path (@rpath) set. In certain cases an attacker can abuse this feature to run arbitrary executable for code execution and privilege escalation. Remove the compiler option -rpath to remove @rpath.</p>	<p>True <a href="#">info</a></p> <p>This binary has a code signature.</p>	<p>True <a href="#">info</a></p> <p>This binary is encrypted.</p>	<p>False <a href="#">warning</a></p> <p>Debug Symbols are available. To strip debugging symbols, set Strip Debug Symbols During Copy to YES, Deployment Postprocessing to YES, and Strip Linked Product to YES in project's build settings.</p>

NO	DYLIB/Framework	NX	STACK CANARY	ARC	RPATH	CODE SIGNATURE	ENCRYPTED	SYMBOLS STRIPPED
28	Payload/S4Cleaner.app/Frameworks/GRDB.framework/GRDB	<p>False <a href="#">info</a></p> <p>The binary does not have NX bit set. NX bit offer protection against exploitation of memory corruption vulnerabilities by marking memory page as non-executable. However iOS never allows an app to execute from writeable memory. You do not need to specifically enable the 'NX bit' because it's always enabled for all third-party code.</p>	<p>True <a href="#">info</a></p> <p>This binary has a stack canary value added to the stack so that it will be overwritten by a stack buffer that overflows the return address. This allows detection of overflows by verifying the integrity of the canary before function return.</p>	<p>True <a href="#">info</a></p> <p>The binary is compiled with Automatic Reference Counting (ARC) flag. ARC is a compiler feature that provides automatic memory management of Objective-C objects and is an exploit mitigation mechanism against memory corruption vulnerabilities.</p>	<p>True <a href="#">warning</a></p> <p>The binary has Runpath Search Path (@rpath) set. In certain cases an attacker can abuse this feature to run arbitrary executable for code execution and privilege escalation. Remove the compiler option -rpath to remove @rpath.</p>	<p>True <a href="#">info</a></p> <p>This binary has a code signature.</p>	<p>True <a href="#">info</a></p> <p>This binary is encrypted.</p>	<p>False <a href="#">warning</a></p> <p>Debug Symbols are available. To strip debugging symbols, set Strip Debug Symbols During Copy to YES, Deployment Postprocessing to YES, and Strip Linked Product to YES in project's build settings.</p>



NO	DYLIB/Framework	NX	STACK CANARY	ARC	RPATH	CODE SIGNATURE	ENCRYPTED	SYMBOLS STRIPPED
29	Payload/S4Cleaner.app/Frameworks/FirebasePerformance.framework/FirebasePerformance	<p>False <a href="#">info</a></p> <p>The binary does not have NX bit set. NX bit offer protection against exploitation of memory corruption vulnerabilities by marking memory page as non-executable. However iOS never allows an app to execute from writeable memory. You do not need to specifically enable the 'NX bit' because it's always enabled for all third-party code.</p>	<p>True <a href="#">info</a></p> <p>This binary has a stack canary value added to the stack so that it will be overwritten by a stack buffer that overflows the return address. This allows detection of overflows by verifying the integrity of the canary before function return.</p>	<p>True <a href="#">info</a></p> <p>The binary is compiled with Automatic Reference Counting (ARC) flag. ARC is a compiler feature that provides automatic memory management of Objective-C objects and is an exploit mitigation mechanism against memory corruption vulnerabilities.</p>	<p>True <a href="#">warning</a></p> <p>The binary has Runpath Search Path (@rpath) set. In certain cases an attacker can abuse this feature to run arbitrary executable for code execution and privilege escalation. Remove the compiler option -rpath to remove @rpath.</p>	<p>True <a href="#">info</a></p> <p>This binary has a code signature.</p>	<p>True <a href="#">info</a></p> <p>This binary is encrypted.</p>	<p>False <a href="#">warning</a></p> <p>Debug Symbols are available. To strip debugging symbols, set Strip Debug Symbols During Copy to YES, Deployment Postprocessing to YES, and Strip Linked Product to YES in project's build settings.</p>

NO	DYLIB/Framework	NX	STACK CANARY	ARC	RPATH	CODE SIGNATURE	ENCRYPTED	SYMBOLS STRIPPED
30	Payload/S4Cleaner.app/Frameworks/FirebaseInstallations.framework/FirebaseInstallations	<p>False <a href="#">info</a></p> <p>The binary does not have NX bit set. NX bit offer protection against exploitation of memory corruption vulnerabilities by marking memory page as non-executable. However iOS never allows an app to execute from writeable memory. You do not need to specifically enable the 'NX bit' because it's always enabled for all third-party code.</p>	<p>True <a href="#">info</a></p> <p>This binary has a stack canary value added to the stack so that it will be overwritten by a stack buffer that overflows the return address. This allows detection of overflows by verifying the integrity of the canary before function return.</p>	<p>True <a href="#">info</a></p> <p>The binary is compiled with Automatic Reference Counting (ARC) flag. ARC is a compiler feature that provides automatic memory management of Objective-C objects and is an exploit mitigation mechanism against memory corruption vulnerabilities.</p>	<p>True <a href="#">warning</a></p> <p>The binary has Runpath Search Path (@rpath) set. In certain cases an attacker can abuse this feature to run arbitrary executable for code execution and privilege escalation. Remove the compiler option -rpath to remove @rpath.</p>	<p>True <a href="#">info</a></p> <p>This binary has a code signature.</p>	<p>True <a href="#">info</a></p> <p>This binary is encrypted.</p>	<p>False <a href="#">warning</a></p> <p>Debug Symbols are available. To strip debugging symbols, set Strip Debug Symbols During Copy to YES, Deployment Postprocessing to YES, and Strip Linked Product to YES in project's build settings.</p>

NO	DYLIB/Framework	NX	STACK CANARY	ARC	RPATH	CODE SIGNATURE	ENCRYPTED	SYMBOLS STRIPPED
31	Payload/S4Cleaner.app/Frameworks/CocoaLumberjack.framework/CocoaLumberjack	<p>False <a href="#">info</a></p> <p>The binary does not have NX bit set. NX bit offer protection against exploitation of memory corruption vulnerabilities by marking memory page as non-executable. However iOS never allows an app to execute from writeable memory. You do not need to specifically enable the 'NX bit' because it's always enabled for all third-party code.</p>	<p>True <a href="#">info</a></p> <p>This binary has a stack canary value added to the stack so that it will be overwritten by a stack buffer that overflows the return address. This allows detection of overflows by verifying the integrity of the canary before function return.</p>	<p>True <a href="#">info</a></p> <p>The binary is compiled with Automatic Reference Counting (ARC) flag. ARC is a compiler feature that provides automatic memory management of Objective-C objects and is an exploit mitigation mechanism against memory corruption vulnerabilities.</p>	<p>True <a href="#">warning</a></p> <p>The binary has Runpath Search Path (@rpath) set. In certain cases an attacker can abuse this feature to run arbitrary executable for code execution and privilege escalation. Remove the compiler option -rpath to remove @rpath.</p>	<p>True <a href="#">info</a></p> <p>This binary has a code signature.</p>	<p>True <a href="#">info</a></p> <p>This binary is encrypted.</p>	<p>False <a href="#">warning</a></p> <p>Debug Symbols are available. To strip debugging symbols, set Strip Debug Symbols During Copy to YES, Deployment Postprocessing to YES, and Strip Linked Product to YES in project's build settings.</p>

NO	DYLIB/Framework	NX	STACK CANARY	ARC	RPATH	CODE SIGNATURE	ENCRYPTED	SYMBOLS STRIPPED
32	Payload/S4Cleaner.app/Frameworks/FBSDKCoreKit.framework/FBSDKCoreKit	<p>False <a href="#">info</a></p> <p>The binary does not have NX bit set. NX bit offer protection against exploitation of memory corruption vulnerabilities by marking memory page as non-executable. However iOS never allows an app to execute from writeable memory. You do not need to specifically enable the 'NX bit' because it's always enabled for all third-party code.</p>	<p>True <a href="#">info</a></p> <p>This binary has a stack canary value added to the stack so that it will be overwritten by a stack buffer that overflows the return address. This allows detection of overflows by verifying the integrity of the canary before function return.</p>	<p>True <a href="#">info</a></p> <p>The binary is compiled with Automatic Reference Counting (ARC) flag. ARC is a compiler feature that provides automatic memory management of Objective-C objects and is an exploit mitigation mechanism against memory corruption vulnerabilities.</p>	<p>True <a href="#">warning</a></p> <p>The binary has Runpath Search Path (@rpath) set. In certain cases an attacker can abuse this feature to run arbitrary executable for code execution and privilege escalation. Remove the compiler option -rpath to remove @rpath.</p>	<p>True <a href="#">info</a></p> <p>This binary has a code signature.</p>	<p>True <a href="#">info</a></p> <p>This binary is encrypted.</p>	<p>False <a href="#">warning</a></p> <p>Debug Symbols are available. To strip debugging symbols, set Strip Debug Symbols During Copy to YES, Deployment Postprocessing to YES, and Strip Linked Product to YES in project's build settings.</p>

NO	DYLIB/Framework	NX	STACK CANARY	ARC	RPATH	CODE SIGNATURE	ENCRYPTED	SYMBOLS STRIPPED
33	Payload/S4Cleaner.app/Frameworks/SwiftyJSON.framework/SwiftyJSON	<p>False <a href="#">info</a></p> <p>The binary does not have NX bit set. NX bit offer protection against exploitation of memory corruption vulnerabilities by marking memory page as non-executable. However iOS never allows an app to execute from writeable memory. You do not need to specifically enable the 'NX bit' because it's always enabled for all third-party code.</p>	<p>True <a href="#">info</a></p> <p>This binary has a stack canary value added to the stack so that it will be overwritten by a stack buffer that overflows the return address. This allows detection of overflows by verifying the integrity of the canary before function return.</p>	<p>True <a href="#">info</a></p> <p>The binary is compiled with Automatic Reference Counting (ARC) flag. ARC is a compiler feature that provides automatic memory management of Objective-C objects and is an exploit mitigation mechanism against memory corruption vulnerabilities.</p>	<p>True <a href="#">warning</a></p> <p>The binary has Runpath Search Path (@rpath) set. In certain cases an attacker can abuse this feature to run arbitrary executable for code execution and privilege escalation. Remove the compiler option -rpath to remove @rpath.</p>	<p>True <a href="#">info</a></p> <p>This binary has a code signature.</p>	<p>True <a href="#">info</a></p> <p>This binary is encrypted.</p>	<p>False <a href="#">warning</a></p> <p>Debug Symbols are available. To strip debugging symbols, set Strip Debug Symbols During Copy to YES, Deployment Postprocessing to YES, and Strip Linked Product to YES in project's build settings.</p>

NO	DYLIB/Framework	NX	STACK CANARY	ARC	RPATH	CODE SIGNATURE	ENCRYPTED	SYMBOLS STRIPPED
34	Payload/S4Cleaner.app/Frameworks/FirebaseCore.framework/FirebaseCore	<p>False <a href="#">info</a></p> <p>The binary does not have NX bit set. NX bit offer protection against exploitation of memory corruption vulnerabilities by marking memory page as non-executable. However iOS never allows an app to execute from writeable memory. You do not need to specifically enable the 'NX bit' because it's always enabled for all third-party code.</p>	<p>True <a href="#">info</a></p> <p>This binary has a stack canary value added to the stack so that it will be overwritten by a stack buffer that overflows the return address. This allows detection of overflows by verifying the integrity of the canary before function return.</p>	<p>True <a href="#">info</a></p> <p>The binary is compiled with Automatic Reference Counting (ARC) flag. ARC is a compiler feature that provides automatic memory management of Objective-C objects and is an exploit mitigation mechanism against memory corruption vulnerabilities.</p>	<p>True <a href="#">warning</a></p> <p>The binary has Runpath Search Path (@rpath) set. In certain cases an attacker can abuse this feature to run arbitrary executable for code execution and privilege escalation. Remove the compiler option -rpath to remove @rpath.</p>	<p>True <a href="#">info</a></p> <p>This binary has a code signature.</p>	<p>True <a href="#">info</a></p> <p>This binary is encrypted.</p>	<p>False <a href="#">warning</a></p> <p>Debug Symbols are available. To strip debugging symbols, set Strip Debug Symbols During Copy to YES, Deployment Postprocessing to YES, and Strip Linked Product to YES in project's build settings.</p>

NO	DYLIB/Framework	NX	STACK CANARY	ARC	RPATH	CODE SIGNATURE	ENCRYPTED	SYMBOLS STRIPPED
35	Payload/S4Cleaner.app/Frameworks/MijickPopups.framework/MijickPopups	<p>False <a href="#">info</a></p> <p>The binary does not have NX bit set. NX bit offer protection against exploitation of memory corruption vulnerabilities by marking memory page as non-executable. However iOS never allows an app to execute from writeable memory. You do not need to specifically enable the 'NX bit' because it's always enabled for all third-party code.</p>	<p>True <a href="#">info</a></p> <p>This binary has a stack canary value added to the stack so that it will be overwritten by a stack buffer that overflows the return address. This allows detection of overflows by verifying the integrity of the canary before function return.</p>	<p>True <a href="#">info</a></p> <p>The binary is compiled with Automatic Reference Counting (ARC) flag. ARC is a compiler feature that provides automatic memory management of Objective-C objects and is an exploit mitigation mechanism against memory corruption vulnerabilities.</p>	<p>True <a href="#">warning</a></p> <p>The binary has Runpath Search Path (@rpath) set. In certain cases an attacker can abuse this feature to run arbitrary executable for code execution and privilege escalation. Remove the compiler option -rpath to remove @rpath.</p>	<p>True <a href="#">info</a></p> <p>This binary has a code signature.</p>	<p>True <a href="#">info</a></p> <p>This binary is encrypted.</p>	<p>False <a href="#">warning</a></p> <p>Debug Symbols are available. To strip debugging symbols, set Strip Debug Symbols During Copy to YES, Deployment Postprocessing to YES, and Strip Linked Product to YES in project's build settings.</p>

NO	DYLIB/Framework	NX	STACK CANARY	ARC	RPATH	CODE SIGNATURE	ENCRYPTED	SYMBOLS STRIPPED
36	Payload/S4Cleaner.app/Frameworks/FMDB.framework/FMDB	<p>False <a href="#">info</a></p> <p>The binary does not have NX bit set. NX bit offer protection against exploitation of memory corruption vulnerabilities by marking memory page as non-executable. However iOS never allows an app to execute from writeable memory. You do not need to specifically enable the 'NX bit' because it's always enabled for all third-party code.</p>	<p>True <a href="#">info</a></p> <p>This binary has a stack canary value added to the stack so that it will be overwritten by a stack buffer that overflows the return address. This allows detection of overflows by verifying the integrity of the canary before function return.</p>	<p>True <a href="#">info</a></p> <p>The binary is compiled with Automatic Reference Counting (ARC) flag. ARC is a compiler feature that provides automatic memory management of Objective-C objects and is an exploit mitigation mechanism against memory corruption vulnerabilities.</p>	<p>True <a href="#">warning</a></p> <p>The binary has Runpath Search Path (@rpath) set. In certain cases an attacker can abuse this feature to run arbitrary executable for code execution and privilege escalation. Remove the compiler option -rpath to remove @rpath.</p>	<p>True <a href="#">info</a></p> <p>This binary has a code signature.</p>	<p>True <a href="#">info</a></p> <p>This binary is encrypted.</p>	<p>False <a href="#">warning</a></p> <p>Debug Symbols are available. To strip debugging symbols, set Strip Debug Symbols During Copy to YES, Deployment Postprocessing to YES, and Strip Linked Product to YES in project's build settings.</p>



NO	DYLIB/Framework	NX	STACK CANARY	ARC	RPATH	CODE SIGNATURE	ENCRYPTED	SYMBOLS STRIPPED
37	Payload/S4Cleaner.app/Frameworks/GoogleUtilities.framework/GoogleUtilities	<p>False <a href="#">info</a></p> <p>The binary does not have NX bit set. NX bit offer protection against exploitation of memory corruption vulnerabilities by marking memory page as non-executable. However iOS never allows an app to execute from writeable memory. You do not need to specifically enable the 'NX bit' because it's always enabled for all third-party code.</p>	<p>True <a href="#">info</a></p> <p>This binary has a stack canary value added to the stack so that it will be overwritten by a stack buffer that overflows the return address. This allows detection of overflows by verifying the integrity of the canary before function return.</p>	<p>True <a href="#">info</a></p> <p>The binary is compiled with Automatic Reference Counting (ARC) flag. ARC is a compiler feature that provides automatic memory management of Objective-C objects and is an exploit mitigation mechanism against memory corruption vulnerabilities.</p>	<p>True <a href="#">warning</a></p> <p>The binary has Runpath Search Path (@rpath) set. In certain cases an attacker can abuse this feature to run arbitrary executable for code execution and privilege escalation. Remove the compiler option -rpath to remove @rpath.</p>	<p>True <a href="#">info</a></p> <p>This binary has a code signature.</p>	<p>True <a href="#">info</a></p> <p>This binary is encrypted.</p>	<p>False <a href="#">warning</a></p> <p>Debug Symbols are available. To strip debugging symbols, set Strip Debug Symbols During Copy to YES, Deployment Postprocessing to YES, and Strip Linked Product to YES in project's build settings.</p>

NO	DYLIB/Framework	NX	STACK CANARY	ARC	RPATH	CODE SIGNATURE	ENCRYPTED	SYMBOLS STRIPPED
38	Payload/S4Cleaner.app/Frameworks/AppsFlyerLib.framework/AppsFlyerLib	<p>False <a href="#">info</a></p> <p>The binary does not have NX bit set. NX bit offer protection against exploitation of memory corruption vulnerabilities by marking memory page as non-executable. However iOS never allows an app to execute from writeable memory. You do not need to specifically enable the 'NX bit' because it's always enabled for all third-party code.</p>	<p>False <a href="#">high</a></p> <p>This binary does not have a stack canary value added to the stack. Stack canaries are used to detect and prevent exploits from overwriting return address. Use the option -fstack-protector-all to enable stack canaries.</p>	<p>False <a href="#">high</a></p> <p>The binary is not compiled with Automatic Reference Counting (ARC) flag. ARC is a compiler feature that provides automatic memory management of Objective-C objects and protects from memory corruption vulnerabilities. Use compiler option -fobjc-arc to enable ARC or set Objective-C Automatic Reference Counting to YES in project configuration.</p>	<p>False <a href="#">info</a></p> <p>The binary does not have Runpath Search Path (@rpath) set.</p>	<p>True <a href="#">info</a></p> <p>This binary has a code signature.</p>	<p>False <a href="#">warning</a></p> <p>This binary is not encrypted.</p>	<p>False <a href="#">warning</a></p> <p>Debug Symbols are available. To strip debugging symbols, set Strip Debug Symbols During Copy to YES, Deployment Postprocessing to YES, and Strip Linked Product to YES in project's build settings.</p>

NO	DYLIB/Framework	NX	STACK CANARY	ARC	RPATH	CODE SIGNATURE	ENCRYPTED	SYMBOLS STRIPPED
39	Payload/S4Cleaner.app/Frameworks/FBSDKCoreKit_Basics.framework/FBSDKCoreKit_Basics	<p>False <a href="#">info</a></p> <p>The binary does not have NX bit set. NX bit offer protection against exploitation of memory corruption vulnerabilities by marking memory page as non-executable. However iOS never allows an app to execute from writeable memory. You do not need to specifically enable the 'NX bit' because it's always enabled for all third-party code.</p>	<p>True <a href="#">info</a></p> <p>This binary has a stack canary value added to the stack so that it will be overwritten by a stack buffer that overflows the return address. This allows detection of overflows by verifying the integrity of the canary before function return.</p>	<p>True <a href="#">info</a></p> <p>The binary is compiled with Automatic Reference Counting (ARC) flag. ARC is a compiler feature that provides automatic memory management of Objective-C objects and is an exploit mitigation mechanism against memory corruption vulnerabilities.</p>	<p>True <a href="#">warning</a></p> <p>The binary has Runpath Search Path (@rpath) set. In certain cases an attacker can abuse this feature to run arbitrary executable for code execution and privilege escalation. Remove the compiler option -rpath to remove @rpath.</p>	<p>True <a href="#">info</a></p> <p>This binary has a code signature.</p>	<p>True <a href="#">info</a></p> <p>This binary is encrypted.</p>	<p>False <a href="#">warning</a></p> <p>Debug Symbols are available. To strip debugging symbols, set Strip Debug Symbols During Copy to YES, Deployment Postprocessing to YES, and Strip Linked Product to YES in project's build settings.</p>

## </> CODE ANALYSIS

NO	ISSUE	SEVERITY	STANDARDS	FILES
----	-------	----------	-----------	-------

## ! OFAC SANCTIONED COUNTRIES

This app may communicate with the following OFAC sanctioned list of countries.

DOMAIN	COUNTRY/REGION
--------	----------------

## DOMAIN MALWARE CHECK

DOMAIN	STATUS	GEOLOCATION
crl.apple.com	ok	<b>IP:</b> 17.253.13.132 <b>Country:</b> United States of America <b>Region:</b> Florida <b>City:</b> Miami <b>Latitude:</b> 25.774269 <b>Longitude:</b> -80.193657 <b>View:</b> <a href="#">Google Map</a>
api-sdk.turboaiclean.com	ok	<b>IP:</b> 47.252.113.195 <b>Country:</b> United States of America <b>Region:</b> California <b>City:</b> San Mateo <b>Latitude:</b> 37.547424 <b>Longitude:</b> -122.330589 <b>View:</b> <a href="#">Google Map</a>
postbacks-app.com	ok	<b>IP:</b> 34.117.147.68 <b>Country:</b> United States of America <b>Region:</b> Missouri <b>City:</b> Kansas City <b>Latitude:</b> 39.099731 <b>Longitude:</b> -94.578568 <b>View:</b> <a href="#">Google Map</a>
ocsp.apple.com	ok	<b>IP:</b> 17.253.13.146 <b>Country:</b> United States of America <b>Region:</b> Florida <b>City:</b> Miami <b>Latitude:</b> 25.774269 <b>Longitude:</b> -80.193657 <b>View:</b> <a href="#">Google Map</a>
www.apple.com	ok	<b>IP:</b> 23.37.124.29 <b>Country:</b> United States of America <b>Region:</b> California <b>City:</b> San Jose <b>Latitude:</b> 37.339390 <b>Longitude:</b> -121.894958 <b>View:</b> <a href="#">Google Map</a>

## EMAILS

EMAIL	FILE
k@x-.t1	

a@o.pxm9 EMAIL t@t.h n	FILE
f@hs.jwz dz@z9n.dg yt@maw-u.p2 zx@ir.tfxb ls@pxynv.9x npg@o.yk5 hb@1o.tpm æ@vldj-2._us ans@ob.nd5 d@ai.tosq k@t.m1o e@t.pwltmb1 g@f.vo ugy@2.qm edvp@-yqu.2s 6@2.yf m@ae.63 9@hkl.un yz-@6uks.tpb b@ij.1z u@tza.pjtb n9@_oq.uckn _@y4dgrs.8yr -@q.vlj0 q h@f9.wvp x@pc.xjxj3b 60dz@7saypi.mx k18@k.xl []@n.vz v@2r.hxp y@-.oh8 u7@cnrt.nuo c@df.şuyypd s[]@f.rm àj@ed.m3[] 9ኃ@ôqy53.h5 -rnv@k.rg g@p5.3x 7.@_x.5ëb0g q2@y.eh zgg1[].2@eu o6@l6gdw.[]l p@e.b2ã c@e.u5 zç@[],pc d@яk.fa 2dmdrn@h.[]' g@t.fivxmvj x@5_ьc9.gl wřwцkc@9.p7 []@sh.s3è eu4j@b.3jm1etibo ojd@_yrh.sm w@r.m_z 2l@f.gfs	S4Cleaner.app/S4Cleaner

EMAIL	FILE
z@pz.f4 -@g.4j	Payload/S4Cleaner.app/Frameworks/Alamofire.framework/Alamofire

## HARDCODED SECRETS

POSSIBLE SECRETS
API_KEY : AlzaSyAYr2LQ8aD8yMGtc6EwhTaarXc1TEMVjvU
FacebookClientToken : e15496ac400f9fbfe8f57ac043ae74f
AppKey : sjFEIZCKdpYmRMsbraSHtprcnDbjyyUe

## SCAN LOGS

Timestamp	Event	Error
2025-01-11 15:05:35	iOS Binary (IPA) Analysis Started	OK
2025-01-11 15:05:35	Generating Hashes	OK
2025-01-11 15:05:35	Extracting IPA	OK
2025-01-11 15:05:35	Unzipping	OK
2025-01-11 15:05:35	iOS File Analysis and Normalization	OK
2025-01-11 15:05:35	iOS Info.plist Analysis Started	OK
2025-01-11 15:05:35	Finding Info.plist in iOS Binary	OK
2025-01-11 15:05:35	Fetching Details from App Store: com.turbocleaner.aiclean	OK

2025-01-11 15:05:35	Failed to get app details	KeyError('sellerUrl')
2025-01-11 15:05:35	Searching for secrets in plist files	OK
2025-01-11 15:05:35	Starting Binary Analysis	OK
2025-01-11 15:05:36	Dumping Classes from the binary	OK
2025-01-11 15:05:36	Running jtool against the binary for dumping classes	OK
2025-01-11 15:05:38	Library Binary Analysis Started	OK
2025-01-11 15:05:38	Framework Binary Analysis Started	OK
2025-01-11 15:05:38	Analyzing Payload/S4Cleaner.app/Frameworks/Kingfisher.framework/Kingfisher	OK
2025-01-11 15:05:38	Analyzing Payload/S4Cleaner.app/Frameworks/GoogleDataTransport.framework/GoogleDataTransport	OK
2025-01-11 15:05:38	Analyzing Payload/S4Cleaner.app/Frameworks/GCDWebServer.framework/GCDWebServer	OK
2025-01-11 15:05:38	Analyzing Payload/S4Cleaner.app/Frameworks/Alamofire.framework/Alamofire	OK
2025-01-11 15:05:38	Analyzing Payload/S4Cleaner.app/Frameworks/FBAEMKit.framework/FBAEMKit	OK
2025-01-11 15:05:39	Analyzing Payload/S4Cleaner.app/Frameworks/FirebaseSessions.framework/FirebaseSessions	OK
2025-01-11 15:05:39	Analyzing Payload/S4Cleaner.app/Frameworks/SwiftUIIntrospect.framework/SwiftUIIntrospect	OK
2025-01-11 15:05:39	Analyzing Payload/S4Cleaner.app/Frameworks/KeychainAccess.framework/KeychainAccess	OK
2025-01-11 15:05:39	Analyzing Payload/S4Cleaner.app/Frameworks/nanopb.framework/nanopb	OK

2025-01-11 15:05:39	Analyzing Payload/S4Cleaner.app/Frameworks/NevSwipeBackGesture.framework/NevSwipeBackGesture	OK
2025-01-11 15:05:39	Analyzing Payload/S4Cleaner.app/Frameworks/CollectionViewPagingLayout.framework/CollectionViewPagingLayout	OK
2025-01-11 15:05:39	Analyzing Payload/S4Cleaner.app/Frameworks/CardStack.framework/CardStack	OK
2025-01-11 15:05:39	Analyzing Payload/S4Cleaner.app/Frameworks/Lottie.framework/Lottie	OK
2025-01-11 15:05:39	Analyzing Payload/S4Cleaner.app/Frameworks/Promises.framework/Promises	OK
2025-01-11 15:05:39	Analyzing Payload/S4Cleaner.app/Frameworks/FirebaseABTesting.framework/FirebaseABTesting	OK
2025-01-11 15:05:39	Analyzing Payload/S4Cleaner.app/Frameworks/ExytePopupView.framework/ExytePopupView	OK
2025-01-11 15:05:39	Analyzing Payload/S4Cleaner.app/Frameworks/JKCategories.framework/JKCategories	OK
2025-01-11 15:05:39	Analyzing Payload/S4Cleaner.app/Frameworks/FirebaseSharedSwift.framework/FirebaseSharedSwift	OK
2025-01-11 15:05:39	Analyzing Payload/S4Cleaner.app/Frameworks/FirebaseCoreInternal.framework/FirebaseCoreInternal	OK
2025-01-11 15:05:39	Analyzing Payload/S4Cleaner.app/Frameworks/FirebaseRemoteConfig.framework/FirebaseRemoteConfig	OK
2025-01-11 15:05:39	Analyzing Payload/S4Cleaner.app/Frameworks/SLImageCompare.framework/SLImageCompare	OK
2025-01-11 15:05:39	Analyzing Payload/S4Cleaner.app/Frameworks/FBLPromises.framework/FBLPromises	OK
2025-01-11 15:05:39	Analyzing Payload/S4Cleaner.app/Frameworks/FirebaseCrashlytics.framework/FirebaseCrashlytics	OK
2025-01-11 15:05:39	Analyzing Payload/S4Cleaner.app/Frameworks/FirebaseRemoteConfigInterop.framework/FirebaseRemoteConfigInterop	OK
2025-01-11 15:05:39	Analyzing Payload/S4Cleaner.app/Frameworks/AppLovinSDK.framework/AppLovinSDK	OK



2025-01-11 15:05:39	Analyzing Payload/S4Cleaner.app/Frameworks/FirebaseCoreExtension.framework/FirebaseCoreExtension	OK
2025-01-11 15:05:39	Analyzing Payload/S4Cleaner.app/Frameworks/SwifterSwift.framework/SwifterSwift	OK
2025-01-11 15:05:39	Analyzing Payload/S4Cleaner.app/Frameworks/GRDB.framework/GRDB	OK
2025-01-11 15:05:39	Analyzing Payload/S4Cleaner.app/Frameworks/FirebasePerformance.framework/FirebasePerformance	OK
2025-01-11 15:05:40	Analyzing Payload/S4Cleaner.app/Frameworks/FirebaseInstallations.framework/FirebaseInstallations	OK
2025-01-11 15:05:40	Analyzing Payload/S4Cleaner.app/Frameworks/CocoaLumberjack.framework/CocoaLumberjack	OK
2025-01-11 15:05:40	Analyzing Payload/S4Cleaner.app/Frameworks/FBSDKCoreKit.framework/FBSDKCoreKit	OK
2025-01-11 15:05:40	Analyzing Payload/S4Cleaner.app/Frameworks/SwiftyJSON.framework/SwiftyJSON	OK
2025-01-11 15:05:40	Analyzing Payload/S4Cleaner.app/Frameworks/FirebaseCore.framework/FirebaseCore	OK
2025-01-11 15:05:40	Analyzing Payload/S4Cleaner.app/Frameworks/MijickPopups.framework/MijickPopups	OK
2025-01-11 15:05:40	Analyzing Payload/S4Cleaner.app/Frameworks/FMDB.framework/FMDB	OK
2025-01-11 15:05:40	Analyzing Payload/S4Cleaner.app/Frameworks/GoogleUtilities.framework/GoogleUtilities	OK
2025-01-11 15:05:40	Analyzing Payload/S4Cleaner.app/Frameworks/AppsFlyerLib.framework/AppsFlyerLib	OK
2025-01-11 15:05:40	Analyzing Payload/S4Cleaner.app/Frameworks/FBSDKCoreKit_Basics.framework/FBSDKCoreKit_Basics	OK
2025-01-11 15:05:40	Extracting String Metadata	OK
2025-01-11 15:05:40	Extracting URL and Email from IPA	OK

2025-01-11 15:05:43	Performing Malware check on extracted domains	OK
2025-01-11 15:05:45	Fetching IPA icon path	OK
2025-01-11 15:05:46	Detecting Trackers from Domains	OK
2025-01-11 15:05:46	Saving to Database	OK

---

**Report Generated by - MobSF v4.2.9**

Mobile Security Framework (MobSF) is an automated, all-in-one mobile application (Android/iOS/Windows) pen-testing, malware analysis and security assessment framework capable of performing static and dynamic analysis.

© 2025 Mobile Security Framework - MobSF | [Ajin Abraham](#) | [OpenSecurity](#).